ECG MONITORING AND DATA ANALYSIS ON MOBILE PLATFORM

HUANG CHENG

(B.Eng. (Hons.), National University of Singapore)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2015

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information, which have been used in this thesis.

This thesis has also not been submitted for any degree in any university previously.

Huang Cheng

12 August 2015

ACKNOWLEDGEMENTS

I would like to express my heartfelt appreciation to my supervisor, Associate Professor Mehul Motani, for his guidance and support during this project. Without his help, encouraging and patience, I would not be able to finish this project on time.

I am also appreciated for my friends, graduate students Huang Aidi and Fu Xiaotian for their generous comments, and suggestions on my work.

I am also grateful for my team members, research engineers Zhou Chongyu, Janaka Jayasuriya and Shashi Raj Singh. They provided generous help and comments on this project.

TABLE OF CONTENT

SUMMARY

In recent years, the fast paced daily life has necessitated a growth in monitoring the health conditions of elderly growing population. Body area network (BAN), showing its overall advantages, is becoming one of the most promising topics, especially targeting the high cardiac disease. In order to provide a conveniently applied solution for detecting heart diseases, Electrocardiogram (ECG) monitoring system for daily life usage is becoming to be a hot topic.

With the booming of the mobile Internet, nowadays, smart phones, or any other smart devices have gradually become the most indispensable part of people's daily life, which therefore, have great potential to build as a portable, smart, stable, and independent ECG monitoring system. In this study, an ECG monitoring system based on the mobile platform is achieved to realize the possibility. We have achieved the following targets.

1. We design and develop an ECG monitoring system based on mobile platform, which contains two modes: gateway mode and server mode to target two different groups of people. Without the support from Internet nor backend server, the system could receive, process, store and display the ECG data in real time independently.
2. The system could collect and process accelerometer data at the same time. We have also collected and analyzed the relationship between the ECG data and the accelerometer data. Furthermore, an abnormal detection algorithm

has been proposed based on the correlation between ECG sensor data and accelerometer sensor data.

LIST OF TABLES

LIST OF FIGURE

LIST OF SYMBOLS

BAN	Body area network
ECG	Electrocardiogram
ORM	Object-relational mapping
OS	Operating system
API	Application programming interface
mHealth	Mobile health
UI	User interface
GPIO	General purpose input output

CHAPTER 1

INTRODUCTION

1.1 Introduction

ECG cardio-monitoring system based on the mobile platforms, which provide effective real time detection of heart condition, while simultaneously satisfies the requirements for a portable and easy accessed body area network (BAN), has attracted great interest for its fundamental properties and potentially practical applications [1]. In this rapid society, one's health condition may become an ignorable element when facing the various stress and workload. Among all the health problems, heart failure, associated with significant morbidity, mortality, and healthcare expenditure, is one of the biggest problems [2]. Hence it is important to gather people's real time ECG information especially for the elders. However, the current medical ECG monitoring system in the hospital is heavy, large, and highly professional, which are impossible to be widely applied in the daily life. A portable ECG monitoring system, which could help people to effectively monitor and analyze their fundamental ECG information, on the contrary, is an essential and pragmatic method.

Since the year of 2007, when Apple Inc. released the first generation of iPhone, the mobile device category was redefined [3]. The computation power of mobile devices is increasing significantly every year. More sensors, bigger RAM, more

powerful CPU are built inside the mobile devices [4]. Mobile devices are more generally used for task solving than conventional desktop computers due to their ever-increased advantages of convenience and functionality. Therefore, with the large number of accessible devices, mobile platforms could result in an individual-based ECG condition monitoring system.

1.2 Literature review

Before this project, there are many research projects introduced by many other people on ECG monitoring topic. Many people have already developed ECG monitoring system on embedded system, Linux distributions, or Windows Mobile platforms

In [5], the authors make use of Qt framework to develop a mHealth application on Android platform. In the system design, the Android device forwards the data received from the sensor to the remote server. The remote backend server will process and display the ECG information. However, there are two drawbacks on this design. On one hand, Internet connection is always required in order to allow the system to run properly, on the other hand, the data received by backend server may not in the real time, since there is unpredictable delay on the mobile Internet connection.

In another paper [6], the author introduces a PDA based application, which could use PDA to monitor the ECG information. The PDA is running based on a Linux distribution. A backend server is built to communicate with the PDA,

store ECG data and display ECG data in real time. Because the PDA cannot store data locally, in order to have persistent data storage, the client side needs a good network connection with the server side. The network issue may constrain the usage of the system. Moreover, there is no ECG data processing in the system. They have not introduced any ECG data processing such as R peak detection in the system.

In [7], the authors implement an ECG monitor system, which uses a gateway to forward data. The gateway can be any smart device, such a computer, PDA, etc. The main purpose of the gateway is to receive data from Data logger and Acquisition System (DAQ) and forward the data to the backend server for persistent storage. Two Android applications are developed for patients and doctors respectively. Patients and doctors can monitor the ECG information in the real time. But all in all, the Android application is not able to show the real time ECG data. Only the historical data would be displayed on the mobile device.

In [8], the authors introduce a system to remotely monitor out-of-hospital patients in real time. By using an embedded mobile unit, the real time ECG data and real time GPS location will be transmitted to the backend server. It requires persistent Internet connection of mobile unit. Moreover, there is no data storage in the mobile device, therefore if the Internet connection failure, the ECG data measured would be lost. The mobile device is only forwarding data to the server.

There is no processing of ECG data in the mobile device, and no real-time feedback to the user.

In [9], the authors implement a very interesting application. They put the sensors on the mobile phone. Instead of collecting ECG data continuously, the system will only collect the ECG information when the users are using the mobile phone.

1.3 Motivation and objectives

This project proposes a system based on mobile platforms, which allows people to carry and use during their daily lives. The system is able to help people to easily and conveniently monitor real time ECG information and detect abnormal ECG signals, whenever and wherever they go. Therefore, the project has the following objectives:

1. The system should run in the mobile devices, and be convenient to use in the daily lives. Meanwhile, the system should maintain a stable real time detector and analyzer of the ECG information, under all circumstances, without manual operations from the users.
2. The system runs without requiring an Internet network connection, and more importantly, the system is independently functioning, even if the backend server is unavailable. Running without Internet network can make the system more robust since we cannot ensure mobile Internet is always

available. After receiving and displaying ECG sensor data, the mobile devices should also be able to store all the data in the devices locally.

3. On the other hand, a backend server of the system is subsidiary for the data storage and integrated analysis. If Internet connection is available, the mobile devices should be able to communicate with the backend server to upload the in device ECG data for permanent storage. The doctors or professionals with authority would access the database and do some deep-level analysis.
4. Furthermore, all the analysis is repeatable as long as the data is stored either remotely or in mobile device. Therefore, a comparative exposition of individual's cardiac condition is accessible to users, as well as an overall analysis of the health status among all users, attribute to the backend server, according to their age groups, length and severity of the abnormal phenomenon.
5. Simultaneously with the real time monitoring, the system is able to instantly process the ECG data without affecting any further data collection, which allows real time feedbacks and suggestions.
6. In this thesis, we are not only focusing on ECG sensor data. Instead an extensible system is proposed, which allows further integration of various sensors. Hence, we investigate the algorithm to process multiple sensor data simultaneously, and is therefore, a much comprehensive study than the single sensor. The more sensors that we attached, the more information we can get.

1.4 Thesis outline

This thesis is divided into seven chapters as listed below.

Chapter one gives a brief introduction on this thesis, including the background of the thesis, literature review as well as the motivation of the project.

Chapter two tells about background information including information of ECG, current mobile platforms, background information of Android as well as the reason why we choose Android.

Chapter three describes the overview of the system. Two modes are proposed in the system, one is called gateway mode, and the other is called server mode. The workflows of both modes would be explained in detail.

Chapter four focuses on the system design of the whole system, such as sensors, database, communication protocols, etc.

Chapter five investigates the data collection, and the algorithm how to process the ECG data.

Chapter six introduces the method to measure accelerometer sensor data as well as the algorithm to analyze both ECG data and accelerometer data together.

Chapter seven, which is the last chapter, gives a summary of the thesis and introduces the future work we may continue on this project.

CHAPTER 2

BACKGROUND INFORMATION

ECG information is one of the most important information regarding people's health and heart condition. In this chapter, background information is presented, such as electrocardiogram (ECG), mobile platforms, and Android.

2.1 Electrocardiogram

Electrocardiogram (ECG) is a record of the electrical potential generated by the heart. It detects small electrical potential changes on the skin due to heart muscles [10].

ECG wave consists of waveform components which indicates electrical events during one heartbeat [11]. These waveforms are labeled P, Q, R, S and T waves. The Figure 2-1 [12] depicts one typical sequence of the ECG.

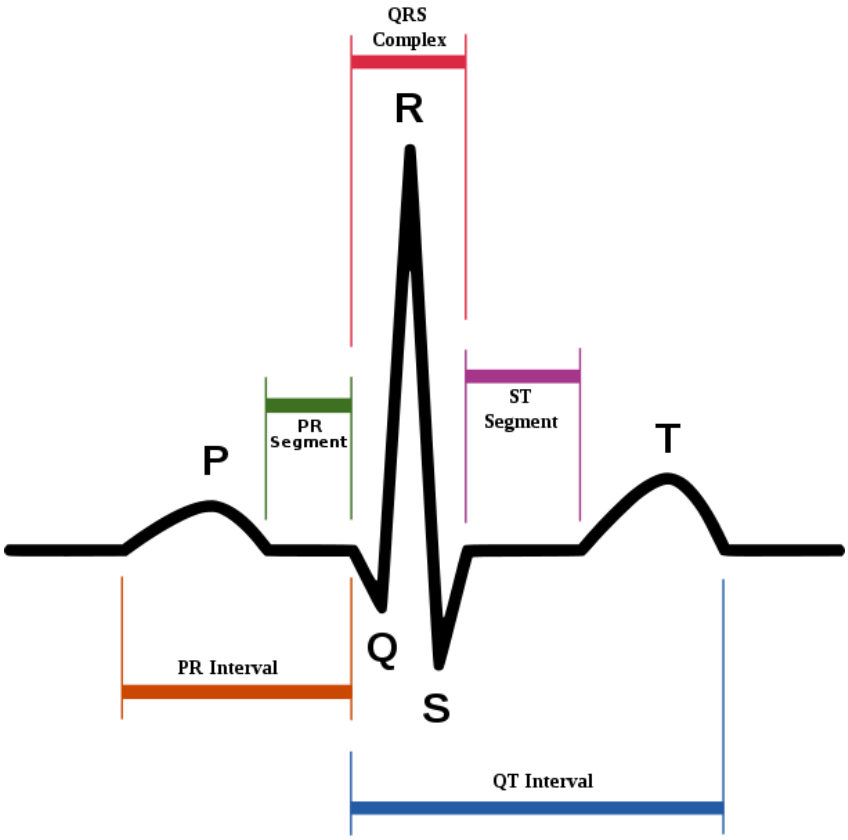

Figure 2-1 ECG of a heartbeat in normal sinus rhythm. [12]

These waveforms indicate the different phases of one heartbeat activity. The P wave represents the normal atrium depolarization. QRS complex represents the ventricular depolarization and contraction. T wave represents the re-polarization of the ventricles [12] [13]. Among all the waveforms, R peak is the most significant waveform. The time interval between the adjacent R peaks is called R-R interval. The heartbeat rate could be calculated by taking the reciprocal of the R-R interval. Hence QRS detection, especially the R peak detection is the most fundamental when processing the ECG data.

The ECG chart and data could provide lot of information. It can help the doctors to know the patients' cardiac condition and evaluate abnormal ECG morphology [14].

2.2 Mobile platforms choice

With the booming mobile Internet, the suitable platforms for the system needs to be settled among all the different mobile platforms. To meet the worldwide requirements, three of them are under consideration: iOS, Android and Windows Phone 8 (WP8).

IOS can be considered to be the OS that opens the era of mobile Internet. It has a high market share, and is well supported by Apple Inc. However, there are many constraints in iOS platform. For example, MFi license [15] is required before we could make use of the hardware, such as transferring data via Bluetooth, in iOS. Applying the MFi may take some time, especially for building up a noncommercial product. Therefore, we give up iOS.

WP8 was an ambitious project by Microsoft. However, nowadays, it takes less than 3 percent mobile market share [16]. In order to allow our system running on more devices, we also give up using WP8.

Compared with other platforms, Android is the best choice. It has a higher market share [16]. It is open source, mostly under Apache license. It grants applications almost any permissions in the mobile devices. Developers are able

to make use of any hardware, any sensors in the mobile phone. It even supports large screen device such as tablet. It also has a good support of multi-tasking. Hence, we decided to develop this system under Android platform.

2.3 Android OS

Android is an open source project, which is led by Google. It can be used on a wide range of different devices, while most of them are mobile devices. [17]. In the year of 2014, Android is taking up more than 75 percent market share of the smart phone operating system [18], therefore it is the largest smart phone operating system in the world.

Since Android is open source mostly under Apache license, which allows anyone and any company to modify and use, it is supported by many companies and organizations and also attracts lots of developers. The programming language in Android is mainly Java, which contains lots of external libraries. In some special applications, which need to do extensive CPU computations, or need to be cross platform, such as graphic games, developers may also embed C/C++ source code in the application.

Android has a good architecture and framework. Java is also a simple and easy language. All these features make Android eco-system healthy and booming. These may be the reasons why Android is growing amazingly [19].

CHAPTER 3

OVERVIEW AND WORKFLOW OF THE SYSTEM

The system is expected to be suitable in different scenarios and situations, therefore, two different modes are designed. One of the modes is called gateway mode, which is used by single person. The other mode is called server mode, which is used to monitor multiple people's information simultaneously.

In this chapter, the basic workflows, features of the system in both two modes are covered. A comparison of the two modes is described at the end of this chapter.

3.1 Workflow of the system

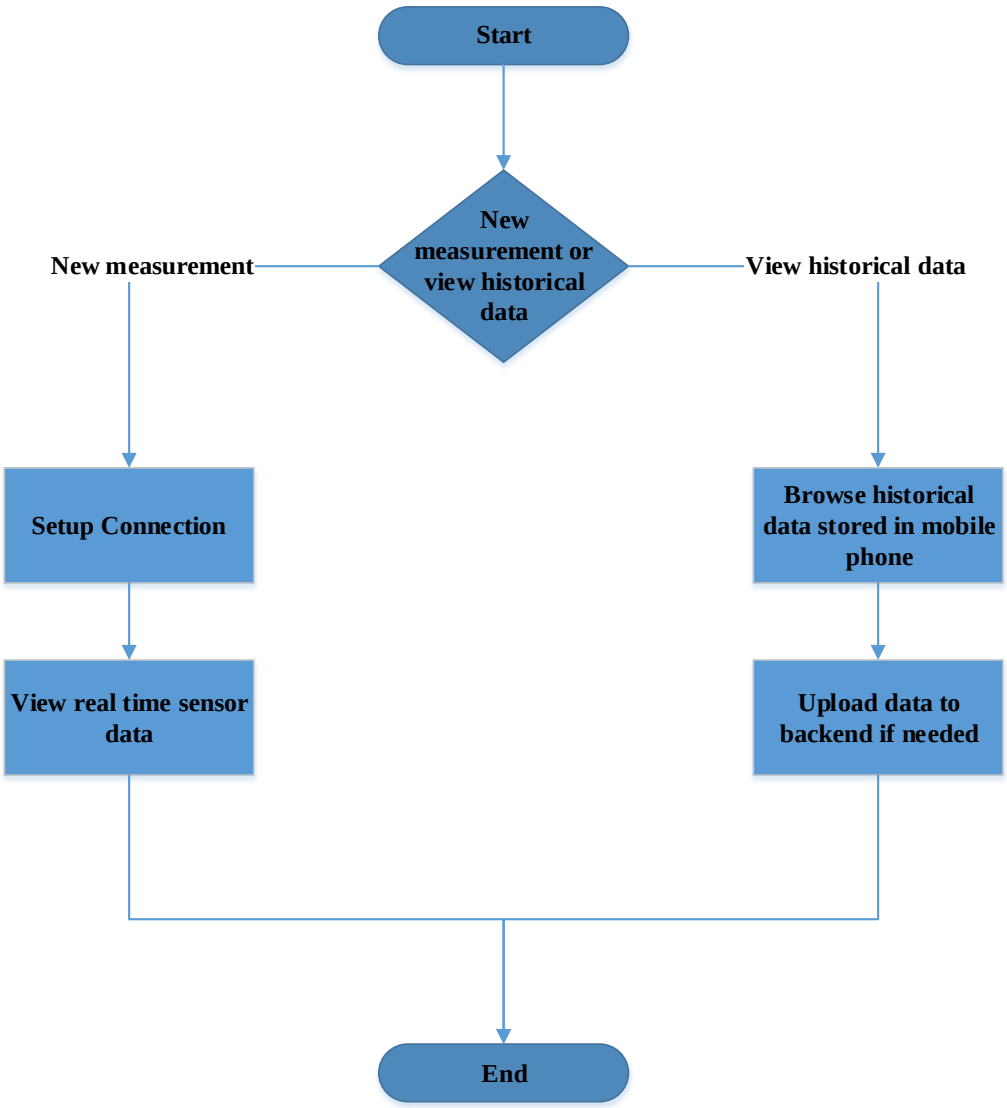

Figure 3-1 Workflow of the system

Figure 3-1 shows a brief workflow of the system for both gateway mode and server mode. The system contains two basic functionalities for both two modes.

1. User can record new session of sensor data using the system. During the recording, the system can detect the sensor data, process the sensor data,

store the data and display the sensor data in real time. After the recording session is finished, the recorded data is stored in local database in the mobile device.

2. User can also browse through the historical data stored in the mobile device. The system also contains a backend server. If Internet network is available, user can choose to upload any historical data to the backend server for deeper analysis in the future.

In next two sections, brief usage scenarios for the two modes are described individually.

3.2 Gateway mode usage scenario

Gateway mode is designed for ordinary people to use during their daily life alone. For instance, in the morning, the user may attach the ECG sensor to his body and launch the mobile device application. Without further manual operation, the system will record down his ECG information for the whole day. Anytime, he could pick up his phone and view his current ECG information such as R peak, heartbeat rate. The whole process does not need any Internet connection or backend server. At the end of the day, the system will store the ECG data for the whole day in the mobile device, and show a general condition of the user's ECG information. Lastly after going back home in the evening, where Wi-Fi connection is available, the user is able to upload the entire ECG data to the backend server easily.

In order to allow normal user to use and monitor, gateway mode is designed to be very lightweight and easy to operate. The hardware of gateway mode is described in Figure 3-2, which mainly comprises three components.

1. ECG sensor detects a person's ECG raw data.
2. Mobile device communicates with the sensor directly to process, store and display the sensor data.
3. Backend server stores the data for deep analysis in the future.

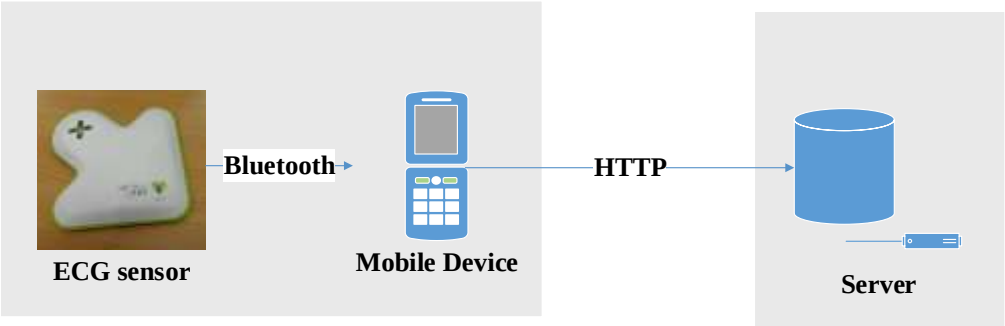

Figure 3-2 Gateway mode scenario

Without the support from the backend server nor Internet, user can carry the sensor and mobile device to record down the ECG data. Gateway mode is easy to use and suitable for one person to self-monitor his ECG condition.

One obvious drawback of gateway mode is that it is not able to monitor multiple users' ECG information simultaneously. In the next section, another mode called server mode will be introduced to solve this drawback.

3.3 Server mode usage scenario

Server mode is designed to monitor sensor data for multiple persons simultaneously. For example, in a small sickroom, or ambulance, the doctors need to monitor ECG information for multiple patients. This is why we design server mode for our system. Its usage scenario is different from gateway mode. The system design of server mode is slightly different and more complex.

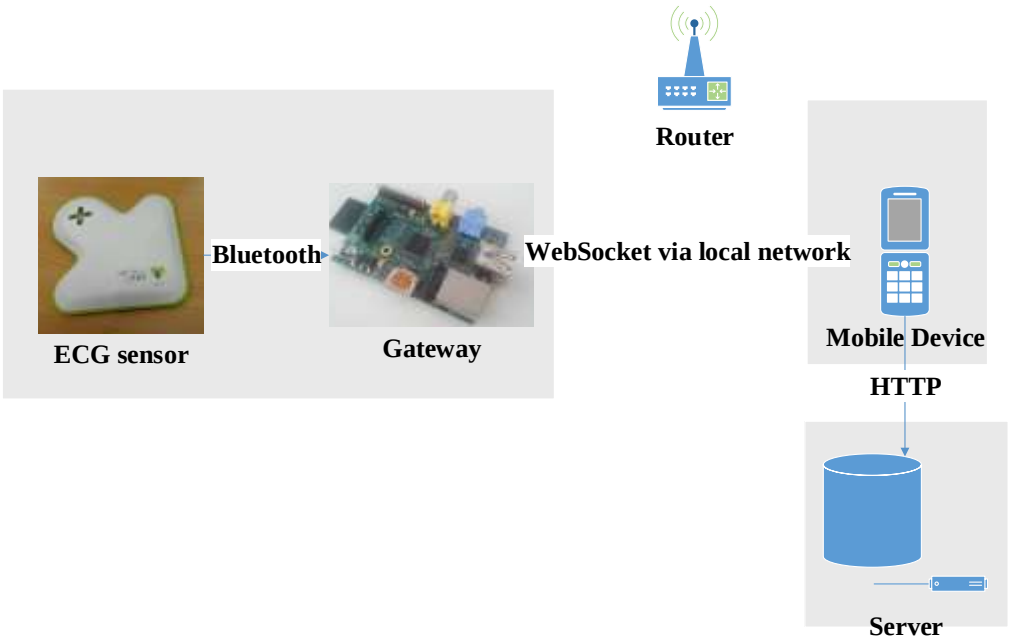

Figure 3-3 Server mode scenario

Figure 3-3 is a brief hardware design of the server mode. There are mainly four components in the server mode.

1. Same ECG sensor is used to detect the raw sensor data.
2. Gateway receives the raw data from the ECG sensor, pre-processes the data and forwards the data to the mobile device.
3. Mobile device is used to receive pre-processed sensor data from the gateway and process, store, display the information to the user. In system design, multiple gateways are able to communicate with one single mobile device.
4. Same backend server is used to store the sensor data uploaded from the mobile devices.

In server mode, the mobile device is able to communicate with multiple gateways simultaneously. Therefore, one mobile device is able to process and display ECG information for multiple persons at the same time.

3.4 Two modes comparison

In the system design, many common modules are reused in the two modes. The only difference between the two modes is that the server mode needs an extra gateway, which helps the mobile device to communicate with the sensors properly.

All the other modules, including, ECG sensor, the data processing and storing logic in the mobile device, the backend server design and communication protocol, are almost the same in both of the two modes. Reusing many common modules can increase productivity and reduce the cost of software development [20].

CHAPTER 4

SYSTEM DESIGN

This chapter introduces system design for both gateway mode and server mode in detail.

4.1 Gateway mode system design

In this section, the system design of the gateway mode is introduced. The Figure 4-1 is an system design diagram of the gateway mode. It contains three parts, ECG sensor, mobile device and backend server.

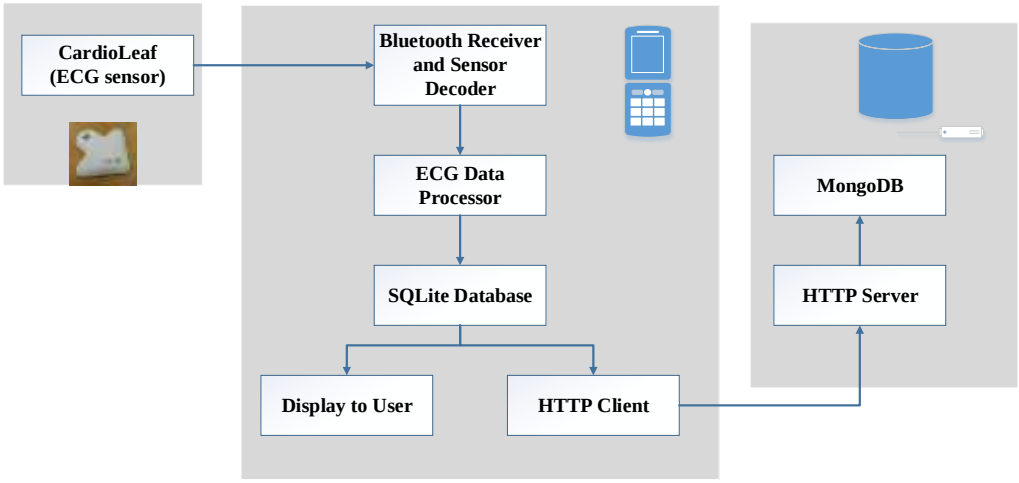

Figure 4-1 Gateway mode architecture

4.1.1 CardioLeaf ECG sensor

The hardware on the left of Figure 4-1 is the ECG sensor, which is used to detect the raw ECG data from user's body. In our system design, we choose

CardioLeaf Fit as the ECG sensor. It is integrated with Bluetooth and able to transmit the encoded ECG data directly to the mobile device via Bluetooth [21]. The data transferred from CardioLeaf is encoded in certain confidential format. It contains 3-Lead ECG recording, with the ultra-low power chip as well as small and thin design [21]. One user can easily carry and use it to monitor ECG data anytime, anywhere.

4.1.2 Mobile device

The hardware in the middle of Figure 4-1 is the mobile device. The mobile device is the core processor of the whole system. All the complicated logic and controls are running in the mobile device. There are several functionalities in the mobile device.

1. Firstly, the mobile device is able to communicate with the ECG sensor. The communication between the ECG sensor and mobile device is via Bluetooth. After receiving the sensor data, the mobile device decodes the raw data.
2. The second functionality of the mobile device is to process the sensor data, including removing baseline wandering and R peak detection on the mobile device. The algorithm of sensor data processing is demonstrated in the next chapter in detail.
3. After processing the data, the mobile device stores the data in the local SQLite database. Consequently, the user can browse through the stored data any time.

4. Fourthly, the mobile device can plot and display ECG chart or other information to the user in real time.
5. All the above functionalities from 1 to 4 do not need Internet connection. The mobile device is able to operate standalone.
6. Lastly, the mobile device is able to communicate with the backend server and store all the local sensor data to the backend server when Internet connection is available.

4.1.3 Backend server

The hardware on the right of Figure 4-1 is the backend server. In order to allow deeper analysis on desktop computer, a backend server is designed and built to store all the sensor data. The backend server is built by my teammates.

1. The backend server is written in Scala programming language with Play framework. Play framework is a high velocity web framework, which supports both Java and Scala language [31].
2. The backend server makes use of MongoDB as the database. MongoDB is a NoSQL database. The key advantage of NoSQL database is that it can store different data storage models with dynamic schemas. Instead of dealing with different rows, developers can store dissimilar data in the database together [32]. New sensor data with different data storage models and dynamic schemas can be stored easily.

3. Sensor data is not necessary to be transferred to the server side in real time. Instead only when the user chooses to upload the sensor data, the system uploads the data to the backend server. HTTP protocol is used to communicate between the mobile device and the backend server. The communication data is in JSON format.

4.2 Server mode system design

In this section, the system design of another mode of the system, server mode, which is mostly designed for monitoring sensor data for multiple users, is introduced in detail. The Figure 4-2 describes the system design of the server mode.

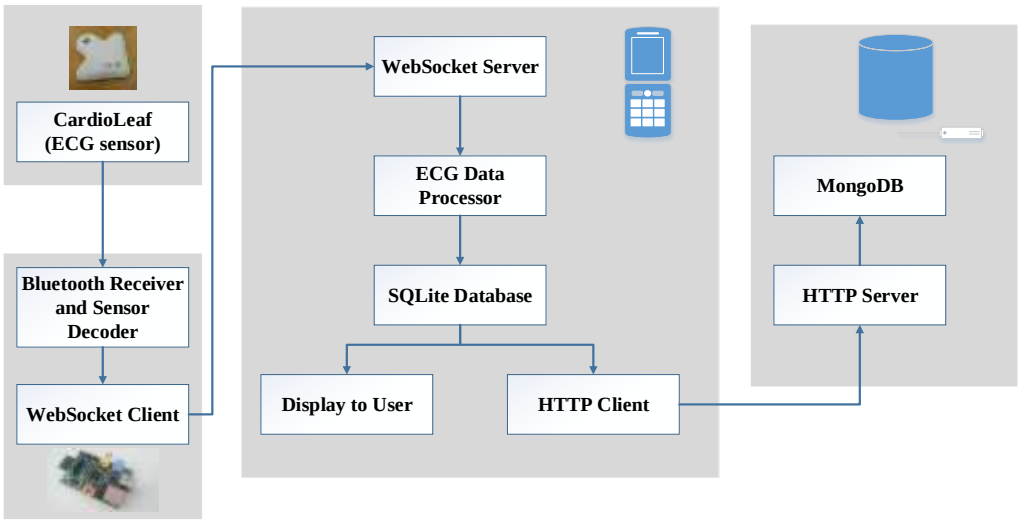

Figure 4-2 Server mode architecture

4.2.1 Similar components from gateway mode

In the server mode, each user holds one sensor and one gateway, while the mobile device is shared by all the users or admin people. The hardware on the top left corner of Figure 4-2 is the ECG sensor. The hardware on the bottom left corner of Figure 4-2 is the gateway. The hardware in the middle is the mobile device. The hardware on the right is the backend server. Except the extra gateway hardware, server mode has almost the same hardware components and similar designs with the gateway mode.

1. The same ECG sensor, CardioLeaf detects raw sensor data
2. The mobile device is still the core processor of the whole system. It contains the same data processing algorithm, and database design.
3. The same backend server is shared in both modes.

The only difference between server mode and gateway mode is that, in the server mode, we design a gateway to help the mobile device to receive sensor data, decode sensor data and forward sensor data. Instead of receiving sensor data directly from the sensor, the mobile device receives the sensor data from the gateway. All the rest architectures and designs are the same between gateway mode and server mode.

Therefore, next section only mentions the unique designs in server mode, which are the gateway hardware design and the communication between the gateway and the mobile device.

4.2.2 Gateway design

In the server mode, the gateway is a message-forwarding node. It communicates with sensors, and forwards the sensor data to the mobile device. Gateway do not need storage, and only need very limited computational power.

Raspberry Pi is chosen to be the gateway. Raspberry Pi is a credit-card sized computer based in ARM architecture, and runs in a customized Linux distribution [22]. It is capable of doing almost everything that a normal Linux computer can do. This means we can integrate various peripherals, as long as the peripherals support Linux.

Software running in the gateway is developed in Python language, since Python can build applications several times faster than Java and C++ languages [23]. It would be a very wise choice than other programming languages.

Connected by a Bluetooth adapter, Raspberry Pi is able to communicate with the CardioLeaf via Bluetooth controlled by the Python application. The communication protocol between CardioLeaf and the gateway is the same in both gateway mode and server mode. We implemented the same decoding algorithm using Python language in the gateway. After decoding the raw data received from CardioLeaf, the gateway forwards the decoded data to the mobile device directly in predefined JSON format.

4.2.3 Gateway and mobile device communication

In the server mode, multiple gateways need to communicate with single mobile device. Since it is difficult to communicate to multiple Bluetooth devices at the same time in Android, WebSocket is chosen as the communication method between the gateways and the mobile device. WebSocket can provide full-duplex communications channels over a single TCP connection. By using WebSocket, the Android device is able to connect to multiple gateways simultaneously [24].

CHAPTER 5

DATA COLLECTION AND PROCESSING

5.1 Original data

After the system is built, the system can be used to collect and display ECG data on the mobile device in real time. For simplicity, the data shown in this chapter is collected using gateway mode.

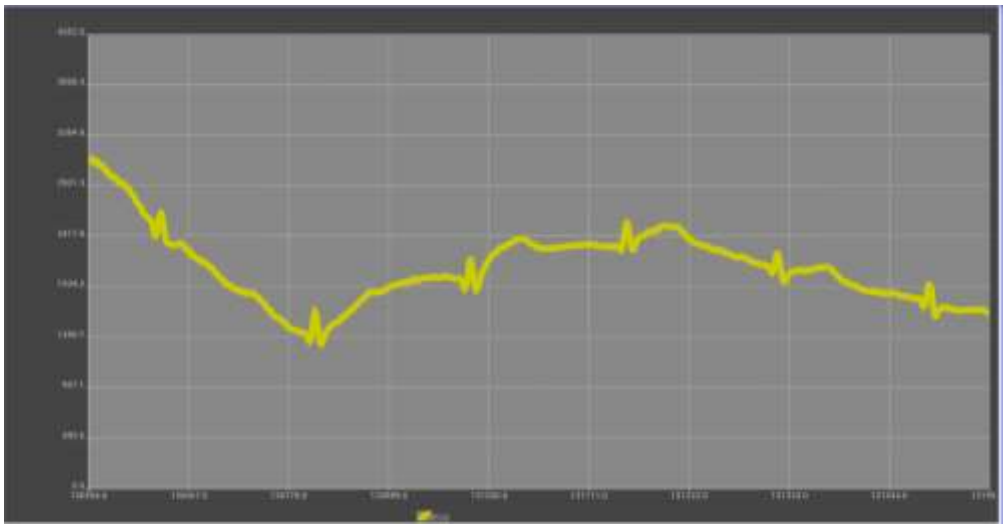

Figure 5-1 Original ECG data

Figure 5-1 shows a sample output of ECG raw data displayed on the mobile device. It is easy to observe that the ECG chart is fluctuating up and down, because the data directly collected from the sensor is affected by the noise. In order to allow the user to view the ECG chart more easily, the raw data need to be processed before displayed on the mobile device.

5.2 Data processing

In this section, data processing algorithms are introduced. There are mainly two processing steps.

1. The first step is removal of baseline wander.
2. The second step is R peak detection in real time on the mobile device.

5.2.1 Removal of baseline wander

The original ECG raw data received from CardioLeaf sensor contains noise. There are many kinds of noise in the raw data, such as power line interface, baseline wandering, electrode contact noise, motion artifacts, and muscle artifacts, etc. [25].

Among all the noise, baseline wander is one of the most common noise. Baseline wandering, which could be considered as a low frequency additional noise to the ECG [26], is the first problem that we need to solve. The fluctuating in Figure 5-2 is caused by the baseline wandering, which causes it difficult to detect R peak.

There are several methods to remove the baseline wander, while moving average based filtering algorithm is implemented in the system. This algorithm is a part of the algorithms in [27]. It is a very simple, but efficient algorithm to remove baseline wander in real time. A brief demonstration is showed in Figure 5-2

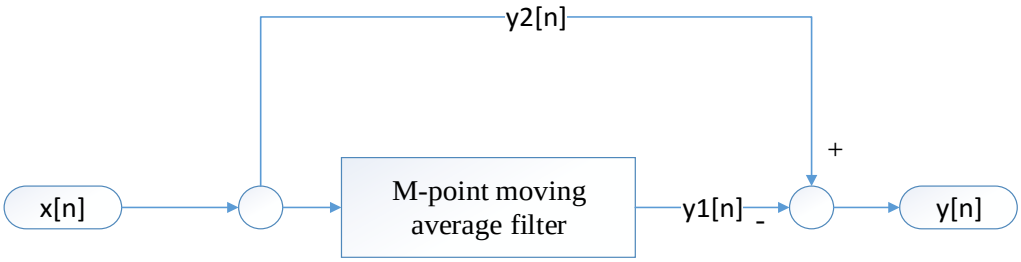

Figure 5-2 Moving average filter algorithm

The detailed algorithm is described below:

1. The x[n] represents the input data, y[n] represents the output data and M is the filter window size, where M must be an odd number.
2. The first output $y_1[n]$ is the average value in the filter window. The relationship between $y_1[n]$ and the input x[n] can be characterized as following:

$$y_1[n] = \frac{1}{M} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} x[n-m]$$

3. The second output $y_2[n]$ is equal to the input x[n].
4. The final system output y[n] can be calculated by subtracting $y_1[n]$ from $y_2[n]$, which could be described as:

$$y[n] = y_2[n] - y_1[n]$$

This filter is able to suppress the lower frequency signals such as P or T waves, as well as the baseline wander [27]. The output and the input data have a group

delay of (M -1) /2, and the algorithm will not able to process the first (M +1)/2 data nor the last (M+1)/2 data. After testing, we choose the window size M to be 31, which results a quite good output.

The original input ECG value $x[n]$ from CardioLeaf ranges from 0 to 4096. This output number is encoded linearly from real voltage value by CardioLeaf. We normalize the output data, $y_{[n]}$, and adjust the average value of normalized output roughly equal to 0.5. Therefore, the following function is used to normalize the output $y_{[n]}$.

$$y_{[n]normalized} = \frac{(y_{[n]} + 4096)}{4096 * 2}$$

After the output data is normalized, the output of processed data is showing in Figure 5-3.

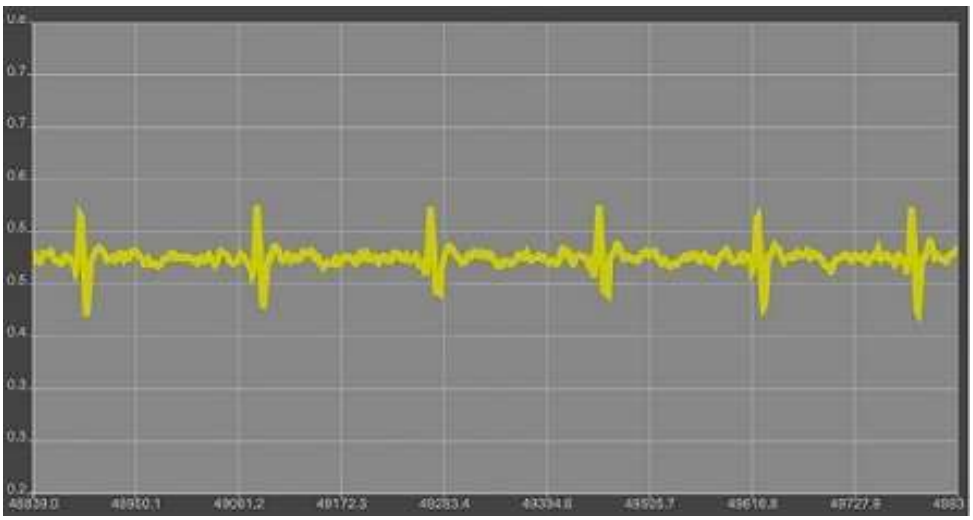

Figure 5-3 Processed and normalized ECG data

Compared with the original data shown in Figure 5-1, the processed data is much more stable without fluctuating.

5.2.2 R peak detection

After removing the baseline wandering, we make use of the processed data to detect R peaks in real time. R peak detection is very essential in our system, since it is one of the most important characteristics in ECG. After detecting R peak, we are able to calculate many properties such as heartbeat rate. The algorithm is implemented in the mobile device in real time with limited computational resources, hence the algorithm should be simple, fast and easy to implement.

After collecting lots of data and analyzing the data in MATLAB, we figure out one suitable R peak detection algorithm. We implemented the algorithm in the mobile device. The algorithm is mainly to detect the shape of the QRS complex,

and therefore find the R peak that fulfil the requirements. This algorithm is also learnt from [27].

1. Firstly, we need to detect the pattern for QRS complex. Q wave and S wave are local minima, while R peak is a local maximum. Therefore, we need to find a pattern that first decrease continuously to a local minimum, which is the Q wave, and then increase continuously to a local maximum, which is the R peak, and then decrease again to a local minimum, which is the S wave, and then increase back to normal.
2. Secondly, we calculate an average value for the past m points, where m is a chosen constant. We define this average value to be μ. Whenever we receive a new ECG data, we will update the average value μ. M just need to be a large enough number, empirically, we choose m to be 3000.
3. After detected such pattern and calculated average value, we continue to check if the detected pattern is valid. We define a threshold for the R peak with respect to the average value defined as $Threshold_R$. In other words, if the detected R peak is larger than $\mu + Threshold_R$, this detected R peak is a valid R peak.
4. Similarly, we apply similar constraint for S wave. We define threshold for S wave with respected to μ as $Threshold_S$. The value of S wave should always smaller than the average value. Therefore, if the detected S wave is smaller than $\mu - Threshold_S$, we accept this S wave is valid.

5. Q wave is less significant than S wave, therefore, we set constraint for Q wave with respect to R peak. We define threshold for Q wave with respect to the R peak as $Threshold_Q$. If the detected Q wave is smaller than $R - Threshold_Q$, we accept this Q wave is valid.
6. Lastly, we make both of the three thresholds to be adaptive. We may update the value of the R peak threshold as following.

   $$Threshold_R = \alpha_R * \gamma_R * (R - average) + (1 - \alpha_R) * Threshold_R$$

   where R is the valid R peak newly detected. α and γ are predefined constants. α can be considered as "forgetting factor" and γ can be considered as weighting factor [27].
7. Similarly, for Q wave or S wave, we update the threshold as following:

   $$Threshold_Q = \alpha_Q * \gamma_Q * (average - Q) + (1 - \alpha_Q) * Threshold_Q$$

   $$Threshold_S = \alpha_S * \gamma_S * (R - S) + (1 - \alpha_S) * Threshold_S$$

   Where Q means the value of the Q wave value newly detected. S refers to the value of S wave newly detected. R is the R peak newly detected.
8. Empirically, we choose the value of α and γ to be the same for all the Q wave, R peak, S wave as follows:

   $$\alpha = 0.2$$

   $$\gamma = 0.4$$
9. Lastly, we set a constraint for the width of the QRS complex. The sampling rate of CardioLeaf is 250Hz. Normal QRS is 70ms to 100ms [28], hence we define the maximum width of one QRS complex to be 25 points. That means

the QRS complex detected should not exceed more than 25 points, otherwise it is not a valid QRS complex.

After implementing this detection algorithm into the system, we collected ECG data using this system when user is seated without moving. 130000 ECG data is collected in more than 8 minutes in one session. The R peak detection performance result is shown in the Table 1.

Total number of R peak	Total Detected	False positive detection	Detection rate
664	661	0	99.5%

Table 1 R peak detection results

The detection rate is about 99.5%. The processed data as well as the peak would be displayed to the user in real time. A screenshot of the chart displayed on the mobile device in real time is shown in Figure 5-4.

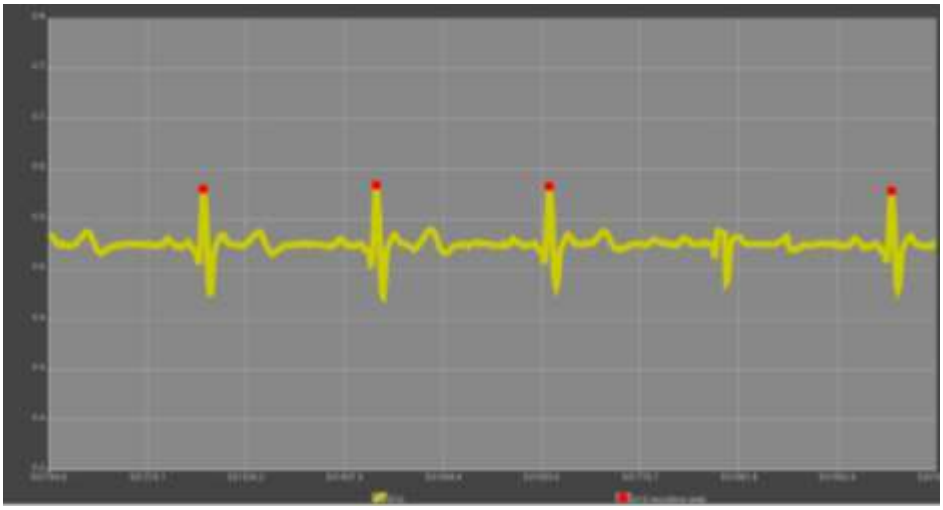

Figure 5-4 ECG with peak detection

The yellow line, is the processed and normalized ECG data, while the red dot is the R peak we have detected. By calculating the R-R interval, we may be able to get the heartbeat rate. We may find that, for most of the time, the algorithm could detect the peak correctly. There is one example of failure to detect the R peak in the Figure 5-4 where the peak become very small unexpectedly. We need investigate and improve the algorithm in the future work.

CHAPTER 6

CORRELATION WITH ACCELEROMETER DATA

As explained in our project objectives, the system is designed to measure different sensors at the same time. In this chapter we explain how we measure and record the accelerometer sensor data and do simple analysis on the relationship between the ECG sensor and the acceleration sensor.

6.1 Accelerometer

Nowadays, various sensors have already been built inside the mobile devices, such as GPS, digital compass, gyroscope. We decided to make use of the sensors inside the mobile device instead of integrating extra sensors into the system. On one hand, it is much easier since we do not need care about the communication between the sensor and the mobile device, on the other hand, it would simplify and lighten the system, because we do not need to integrate extra sensors and modules.

We wanted to collect sensor data inside the phone which has correlation with the ECG information, consequently, we can investigate both sensors to develop an algorithm to detect abnormal signal.

Doing exercise would cause heartbeat increasing, therefore, the sensor need be able to indicate how hard the user is doing exercise. Initially, we thought location sensors such as GPS can indicate how fast the people are running,

therefore could indirectly tell us the intensity of the exercise. However, there are several drawbacks on location service.

1. Location sensors may not accurate especially in indoor environment.
2. Location service is not highly related to how hard the user is doing sports. Many sports such as basketball or running on treadmill do not cause the change of the location.

After analyzing and comparing several sensors, we decided to record the data from accelerometer sensor. Accelerometer can measure the acceleration of the device. When a person is doing exercise with the mobile device, the accelerometer can record down the acceleration at that time accordingly. In the next few sections, we introduce how we measure and process the accelerometer sensor data, conduct experiment and prove that the accelerometer sensor data do have correlation with the ECG sensor data.

6.2 Acceleration index

We need to find an algorithm to calculate how hard the user is doing exercise based on accelerometer sensor data. We develop a simple but useful algorithm to calculate a variable to represent how hard the user is doing exercise. We call the variable to be acceleration index, which can be found highly related to people's exercise. The algorithm to calculate the acceleration index is explained below:

1. In Android, the application can record down acceleration of the phone in three directions, demonstrated in Figure 6-1 [29]. However, the acceleration value returned by the system is always including the gravity acceleration. Hence, in the first step we need to calculate the pure acceleration without the gravity. This could be done by using a high pass filter to the measured data, we have implemented a simple high pass filter according to [30]. The advantage of this high pass filter is that it is simple and need less computational resources, which is suitable for a mobile device. However, one disadvantage is that, it need the previous measured acceleration data to process the current acceleration data. And at the beginning of each recording session, we do not have the previous measured acceleration, therefore, at the very beginning, the processed data is not trustable, but slowly, it will become valid and stable. We may find this phenomenon in Figure 6-2 and Figure 6-3 in the next section. The acceleration without gravity without gravity in three directions are denoted by a_x a_y, a_z respectively.

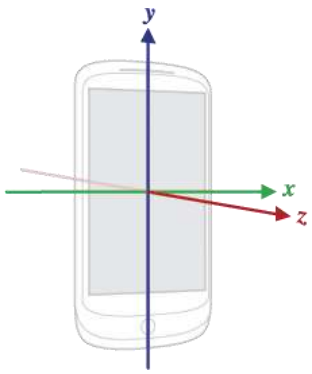

Figure 6-1 Android coordinate system [29]

2. After measuring the acceleration in three directions without gravity, we calculate the vector magnitude of the sum of the acceleration by

$$a_{\text{total}} = \sqrt{a_x * a_x + a_y * a_y + a_z * a_z}$$

3. We sum up the acceleration magnitude in constant time interval. We call the sum to be the acceleration index during that interval. The acceleration index in that interval may explicitly tell us how hard the user was doing exercise.

6.3 System implementation

We need to store all the accelerometer data collected from the mobile device. Hence, we add one more table into the mobile device database to store the accelerometer sensor data. We reuse similar programming structure, and user can view the real time or historical accelerometer sensor data. In Android, the sampling rate for acceleration index sensor is adjustable, and we choose the sampling rate to be 5 Hz. We save the acceleration index into the database every 2 seconds, and display to the user in real time.

After implementing the new features for the system, we have done two simple measurements for the acceleration index. First time, we carried the mobile device and walked slowly in the room. The acceleration index chart is shown in Figure 6-2. The second time, we hold the mobile device and seated on the chair without moving. The acceleration index output chart is shown in Figure 6-3.

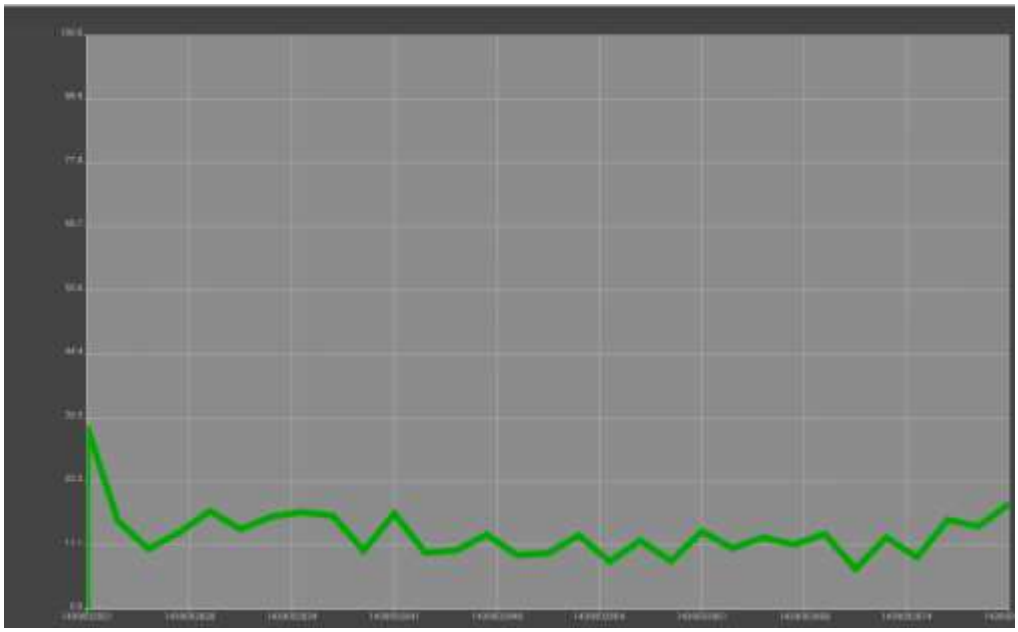

Figure 6-2 Acceleration Index chart when walking

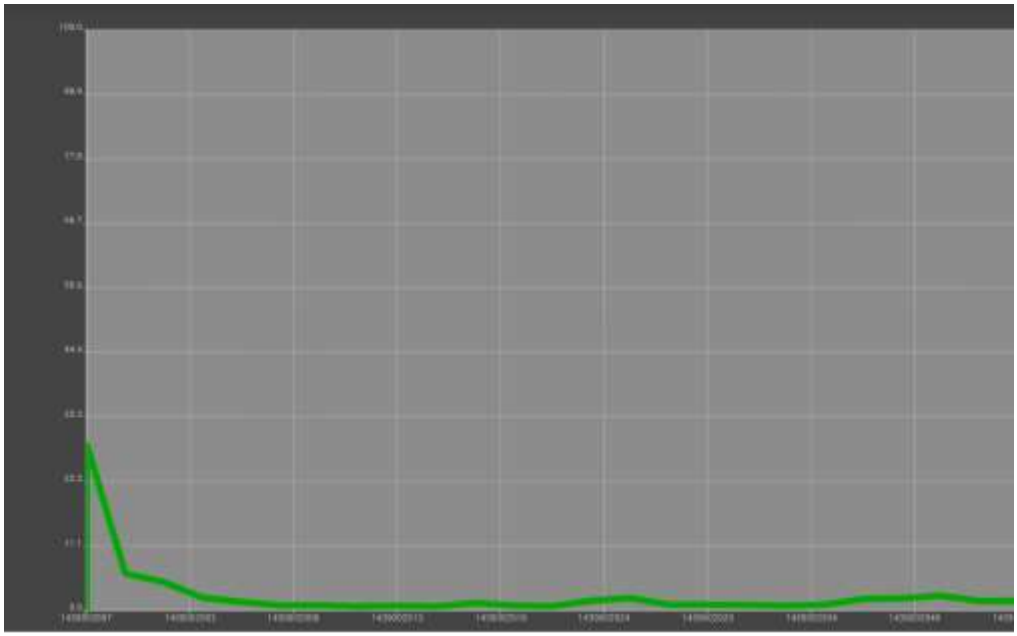

Figure 6-3 Acceleration Index chart when seated on the chair

It can be observed from the chart that when walking, we get a higher acceleration index comparing to when we are seated in the chair. This means the acceleration index has a positive relationship to the exercise intensity. In the next section, we describe how we conduct experiment to collect and analyze the ECG sensor data as well as accelerometer data.

6.4 Experiment on ECG and acceleration index correlation

The exercise intensity is related to the acceleration index and the heartbeat rate. In this section, a brief relationship between acceleration index and the heartbeat rate is investigated, which is helpful to learn abnormal detection in our system.

For example, when a person is doing exercise, both his heartbeat rate and acceleration index may increase significantly. If we only consider the ECG sensor information, we may consider the ECG sensor data is abnormal, since heartbeat increase with no reason. However, if we analyze ECG data and accelerometer data together, we may observe the acceleration index is increasing as well, which means the user is doing exercise. Hence it is acceptable to have an increasing heartbeat rate.

In this section, we conduct an experiment to collect data for both ECG data and acceleration data. On one hand, we use the experiment to test our system, on the other hand, we make use of this experiment to analyze the relationship between the ECG data and the accelerometer data.

MATLAB and Excel are the tools to analyze and plot the sensor data. In next few sections, how we coordinate the experiment and how we collect and analyze the sensor data is explained in detail.

6.4.1 Data collection

We learnt from [31], and implemented a similar experiment to collect ECG data and acceleration data when user is running. We chose to collect the data when we were running on the treadmill, with the same mobile device. I was the only person to run on the treadmill. I had no disease history and I was very healthy when we were conducting this experiment.

We chose 5 different levels of speeds, 2km/h, 4km/h, 6km/h, 8km/h, and 10km/h. For each level of speed, I ran on the treadmill for 5 minutes, then I stopped running and took a rest for another 5 to 8 minutes, in order to allow the heartbeat to become peace. We monitored the ECG information as well as acceleration information during the whole process using our own system.

6.4.2 Overview of the data of ECG and acceleration index

After finishing the data collection, we can analyze the relationship between the acceleration index and ECG sensor data. We exported the data, and process ECG sensor data and the acceleration index as following:

1. In different scenarios, we have different detection rate. In order to analyze the real results, we manually counted the R peak numbers, and calculated

the real ECG heartbeat rate by reading from the chart. We calculate the heartbeat rate every 8 seconds by taking the average value of the past 16 seconds.

2. We also did similar process for the acceleration index. We calculate the acceleration index every 8 seconds by taking the average value for the past 8 seconds.
3. In order to compare the results for the same circumstance, we plotted the heartbeat rate and acceleration index in the same chart for the same scenario using Excel. Figure 6-4 to Figure 6-8 describe the charts in the 5 situations.

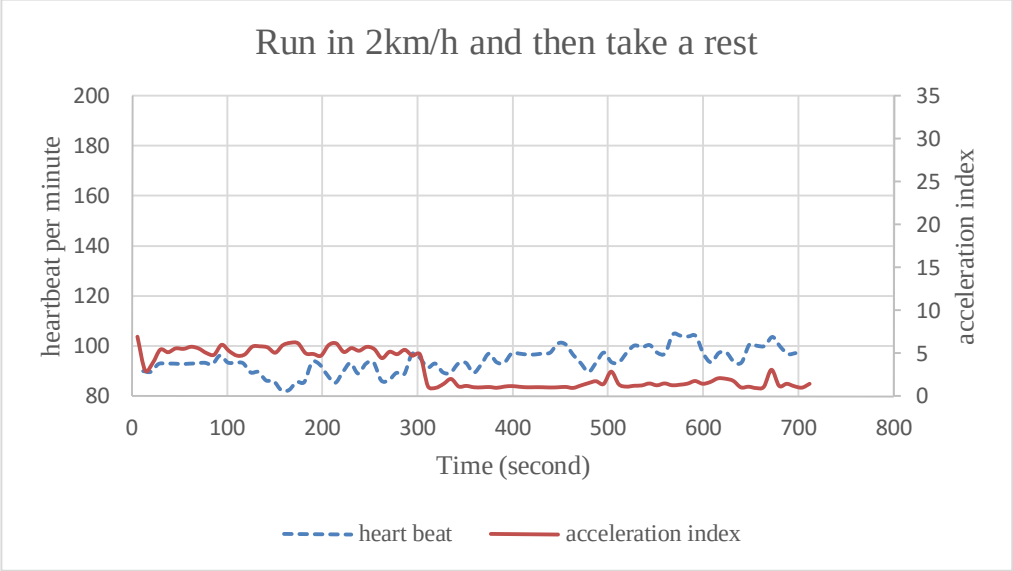

Figure 6-4 Run on the treadmill with a speed of 2km/h for about 300 seconds, and then stop running and take a rest for another 300 seconds. The charts shows the heartbeat rate and acceleration index with respect to the time in the whole process.

Figure 6-5 Run on the treadmill with a speed of 4km/h for about 300 seconds, and then stop running and take a rest for about another 300 seconds. The charts shows the heartbeat rate and acceleration index with respect to the time in the whole process.

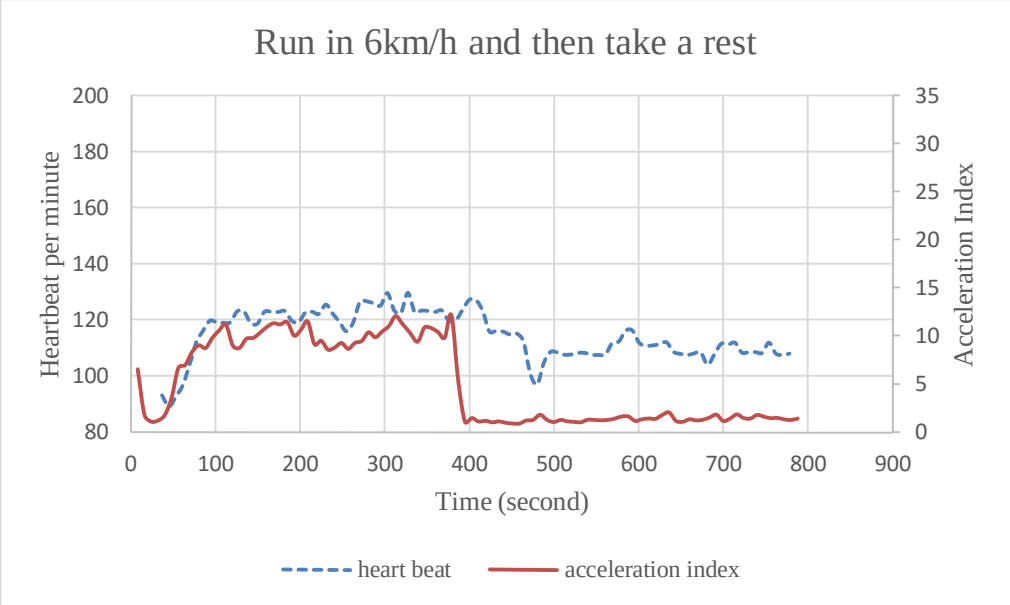

Figure 6-6 Run on the treadmill with a speed of 6km/h for about 300 seconds, and then stop running and take a rest for about 300 seconds. The charts shows the heartbeat rate and acceleration index with respect to the time in the whole process.

Figure 6-7 Run on the treadmill with a speed of 8km/h for about 300 seconds, and then stop running and take a rest for about another 300 seconds. The charts shows the heartbeat rate and acceleration index with respect to the time in the whole process.

Figure 6-8 Run on the treadmill with a speed of 10km/h for about 300 seconds, and then stop running and take a rest for about 300 seconds. The charts shows the heartbeat rate and acceleration index with respect to the time in the whole process.

The charts above describe the changes of the heartbeat rate and the acceleration index with respect to time:

1. The dotted line indicates the heartbeat rate, which based on the scale on the left of the chart. The solid line indicates the acceleration index, which based on the scale on the right. The horizontal axis indicates the time in the unit of second.
2. In order to view and analyze the data in all the scenarios, same vertical axis scale is used in all the charts.

We have several observations based on the charts we plot:

1. We can easily find that all the acceleration indexes in the charts show a very clear cut cliff, which divided the charts into two halves. In the first half, the acceleration index fluctuates along a higher value than the second half. This is because I am running during the first half, while I suddenly stop during the second half.
2. Comparing first half of the 5 cases, we can observe that when we are running in a higher velocity, the acceleration index fluctuates along a higher value.
3. Regarding to the heartbeat rate, we can observe similar results with [31], where we get 3 phases in the heartbeat chart. Initially, when people starts running, the heartbeat rate increases gradually. After increases for some time, the heartbeat rate reaches to a stable maximum value, and keeps fluctuate along the maximum value. Higher running speed results to a higher maximum value. After the people stops running, the heartbeat rate continues dropping to the normal heartbeat rate.
4. In Figure 6-4 and Figure 6-5, we observe that after people stops running, the heartbeat rate even increases a little bit, which is a little strange. But it can be explained according to [32], which is because we stop exercising abruptly. During the exercise, people's heart has been pumping blood and oxygen to supply muscles. On the other hand, the muscles also contract and expand to push blood back to heart. If we stop exercising abruptly, the muscles stop pumping, and the heart must increase its pace to compensate for the work the muscles have been doing. That's why the heartbeat rate increases a little after we have stopped running.

6.4.3 Acceleration index and R peak detection rate

In this section, we discuss the R peak detection rate under different acceleration index.

After the experiment, we have counted how many R peak the system is correctly detected as well as the total R peaks in the whole experiments. We have calculated the detection rate in all scenarios. For each scenario, we have two parts, the first half is the running part, in which we were running in different speed. The second half is the resting part, in which we stopped running, and kept stationary. The data for the detection rate is shown in Table 2 and Table 3.

Scenarios	R peak detection rate
When running at the speed of 2km/h	97.1%
When running at the speed of 4km/h	98.1%
When running at speed of 6km/h	95.2%
When running at speed of 8km/h	91.5%
When running at speed of 10km/h	75.2%

Table 2 R peak detection rate when the user is running in various running speed

Scenarios	R peak detection rate
After stopping running at speed of 2km/h	95.8%
After stopping running at speed of 4km/h	98.1%
After stopping running at speed of 6km/h	96.6%
After stopping running at speed of 8km/h	94.5%
After stopping running at speed of 10km/h	98.3%

Table 3 R peak detection rate when the user is resting after finished running in various speed

We can observe that when the user is stationary, the detection rate is quite stable and acceptable. However, when the user is running, the detection rate become unstable, and decreases with respect to the speed of running.

Based on the data in Table 2 and Table 3, we are able to plot the chart to display the R peak detection rate with respect to the running velocity shown in the Figure 6-9.

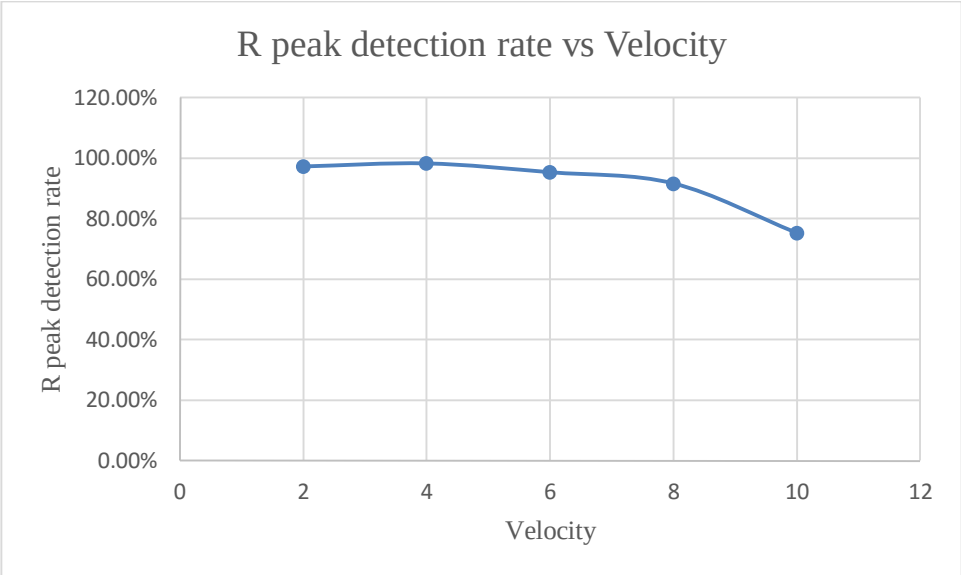

Figure 6-9 R peak detection rate vs velocity

The chart clearly shows how the R peak detection rate related to the running speed. The detection rate drops when the running speed increases, especially when the running velocity is higher than 8km/h.

We investigated the reason why the detection rate drops according to the velocity. The key reason is that, higher velocity would cause a bad contact between the sensor and the body, which would cause noise when receiving the ECG raw data. We have taken two screenshots of the ECG signal charts when running in low velocity and in high velocity shown in Figure 6-10 and Figure 6-11 respectively.

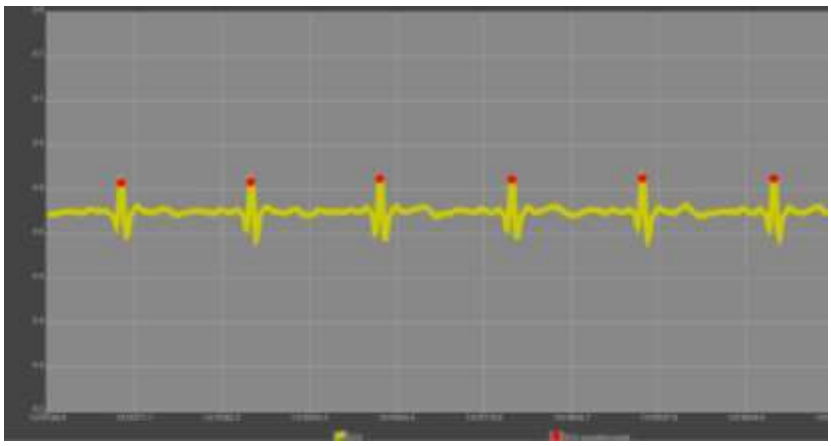

Figure 6-10 Example of ECG signal when running in low velocity

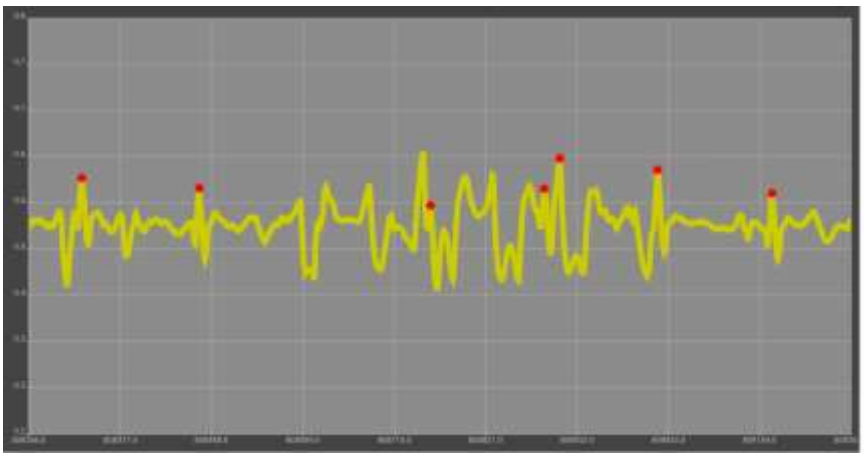

Figure 6-11 Example of ECG signal when running in high velocity

The dot is the peak auto detected by the system. Most of the time, when the detected raw sensor data is valid, the system is able to detect R peak. However, in Figure 6-11, we can observe that when running in a higher velocity, the chart fluctuates more severely and in some extreme cases the data is corrupted due to a higher noise, which causes the system to detect the peak wrongly. Both false detection and missing detection happen. This is mainly due to electrode contact noise. Compared to the normal electrode which is small and light, the

CardioLeaf is a little big and heavy. We have to use tape to stick the CardioLeaf onto the body tightly. However, when the user is running very fast, which causes the body to jump up and down, CardioLeaf may become loose from contacting the body. Therefore, the faster the user runs, the more significant the electrode contact noise will be. We should learn from this phenomenon and improve the detection algorithm, especially when we come across high noise in the ECG data.

6.4.4 Acceleration index and speed

In this section we analyze the relationship between the acceleration index and the running speed. When we are running in a constant speed, the acceleration index fluctuates along a constant value.

We have collected the data between this constant acceleration index and the running speed in different scenarios. Moreover, we also calculated the average value of the acceleration index when the user is rest. The results are shown in the Table 4.

Scenario	Average Acceleration Index
Average value when user is resting, which means running speed is 0km/h	1.422
When running at the speed of 2km/h	5.28
When running at the speed of 4km/h	7.61
When running at the speed of 6km/h	8.88
When running at the speed of 8km/h	14.5
When running at the speed of 10km/h	22.23

Table 4 Acceleration index vs velocity

In order to view the result in an easier way, we plot the acceleration index with respect to velocity in a line chart shown in Figure 6-12 below.

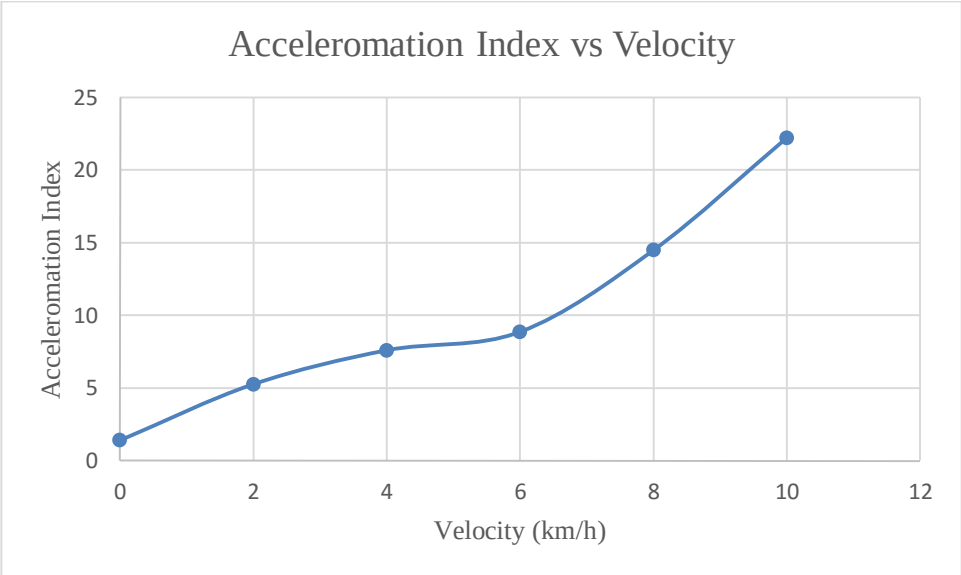

Figure 6-12 Chart of acceleromation index and vecloty

We have several observations from the chart and the table

1. We can notice the acceleration index have a positive relationship with respect to the velocity. When the running speed increases, the acceleration increases as well. Acceleration index does indicate how hard the user is doing exercise.
2. The acceleration index and velocity is not in a perfect linear relationship. When speed is smaller than 4 km/h, the chart is a roughly linear relationship, while speed larger than 6 km/h, the chart is in another roughly linear relationship with a higher gradient.

6.4.5 Maximum stable heartbeat rate and average acceleration index

From the Figure 6-4 to Figure 6-8, we can observe that when user is running, the heartbeat rate increases accordingly, and then keeps almost steady at a

constant stable heartbeat rate. In this section, we analyze the relationship between this stable heartbeat rate and the acceleration index. The data is collected and shown below:

Scenarios	Acceleration index	Stable heartbeat rate
When running at the speed of 2km/h	5.28	90.4
When running at the speed of 4km/h	7.61	112
When running at the speed of 6km/h	8.88	123
When running at the speed of 8km/h	14.5	180
When running at the speed of 10km/h	22.23	183

Table 5 Stable heartbeat rate with respect to the acceleration index under different scenarios

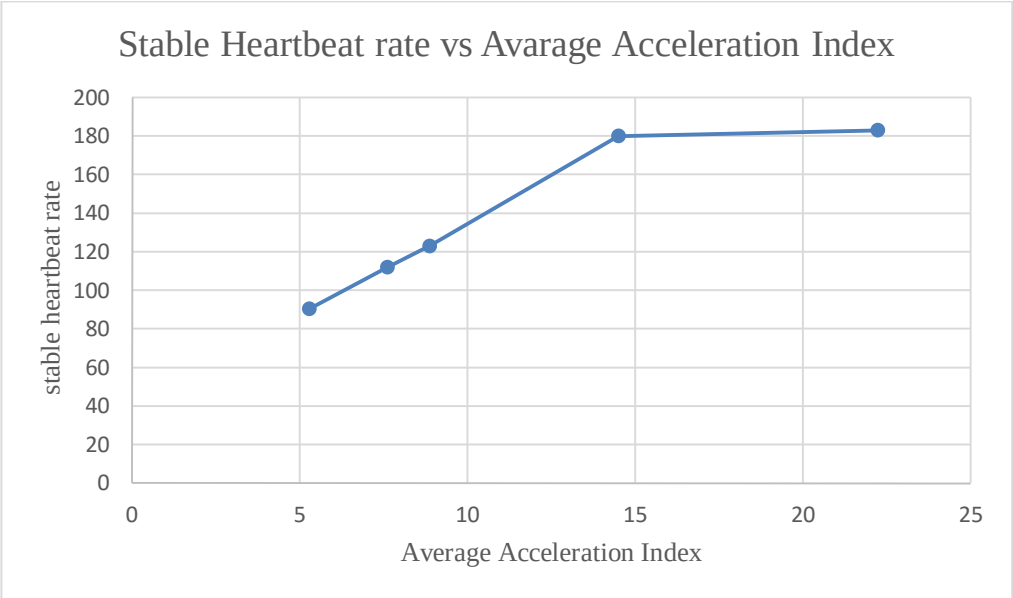

Figure 6-13 Stable heartbeat rate vs acceleration index

We can observe that the stable heartbeat rate increases with respect to the acceleration index until it reaches to a maximum value.

Before reaching to the maximum value, the stable heartbeat rate has a roughly linear relationship with respect to the acceleration index.

6.4.6 Summary of the data collection

In conclusion, we have summarized several characteristics between ECG data and accelerometer data.

Based on the analysis, we have figured out that ECG sensor data does have certain correlation with accelerometer sensor data. Based on the relationship, we can build an algorithm to analysis the sensor data and find a way to detect the abnormal of the user's health information.

6.5 Alert detection

In the previous sections, we analyze the relationship between the acceleration index and the heartbeat rate. Based on the characteristics, we can make use of the sensor data together to develop an algorithm to give user alert if we detected abnormal sensor data.

Recall the chart of acceleration index and heartbeat rate when running in the speed of 8km/h and then taking rest, shown in Figure 6-14. It shows the heartbeat rate and acceleration index changes with respect to time

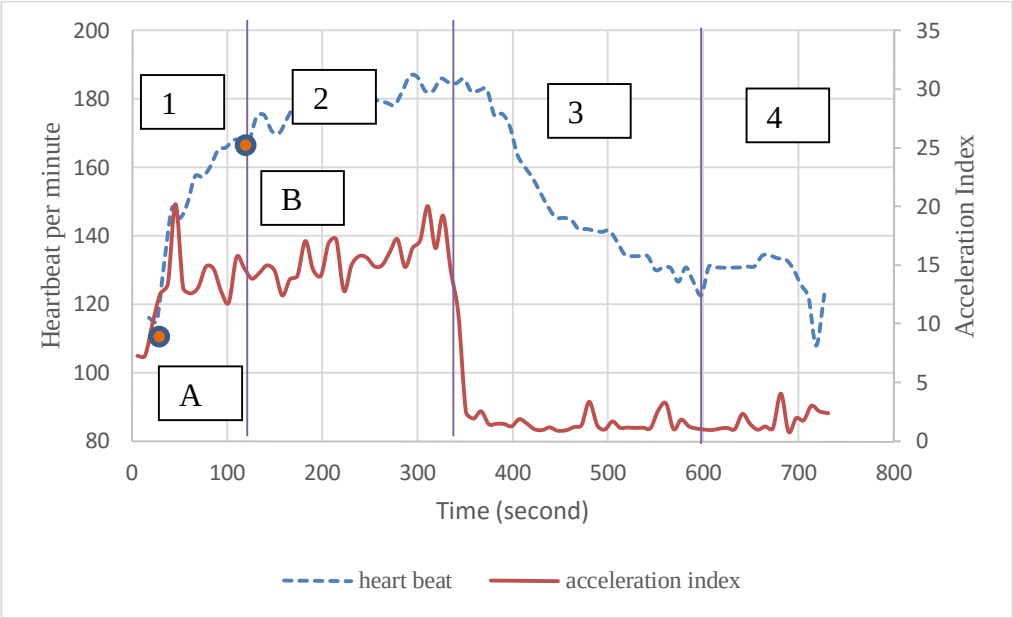

Figure 6-14 Three stages when doing exercise

This chart shows a similar result with [30]. From the chart above, we can find out that when we are doing exercise, we come across three different stages in the chart.

1. The first stage is called the rising stage, in which the heartbeat rate increases significantly due to the user is exercising.
2. The second stage is called stable stage. The heartbeat rate become much stable in this stage. Though the heartbeat rate would still increase slightly, the increasing speed is not very high. Moreover, the heartbeat starts to fluctuate along a constant value.
3. The third stage is called dropping stage. When the user is taking a rest, or exercising with a lower acceleration index, the heartbeat rate decreases significantly with respect to time.

4. The last part is the stable stage again. After some time, when the user gets used to the lower acceleration index, and the heartbeat rate become steady and fluctuating along a constant value again.

6.5.1 General idea

When user is exercising, the heartbeat rate follows the rising, stable and dropping phase over time. Essentially, we need to detect how closely the heartbeat rate follows rising, stable and dropping phases over time as we have observed. If the measured sensor data does not follow our expectation, we can suspect there is something abnormal happening. We have several thoughts and assumption regarding the data above.

1. If the user is doing sports in a constant acceleration index, his heartbeat rate changes gradually, until the heartbeat rate reaching a steady value. The heartbeat rate is highly related to the acceleration index. Actually, in section 6.4.5, we have discussed that this steady value is roughly in a linear relationship with respect to the acceleration index.
2. If previously the user has a lower heartbeat rate than his steady heartbeat rate, after he starts to do exercise in a constant acceleration index, his heartbeat rate would increase gradually until reaching the steady heartbeat rate. In this case, it is the rising stage.

3. If previously, the user has a higher heartbeat rate than his steady heartbeat rate, his heartbeat rate would decrease gradually until reach the steady heartbeat rate. In this case it is the dropping stage.

As we have observed, these three stages have very different characteristics. We define different algorithms for different stages. Therefore, our general algorithm is described as following.

1. We need distinguish these three stages. Firstly, we focus on the stable stage. In stable stage, the heartbeat rate does not change too much. For a given acceleration index, we define the minimum value and maximum value of the heartbeat rate in the stable stage as $HR_{stable-min}$ and $HR_{stable-max}$ respectively. In other words, if the heartbeat rate is within interval $[HR_{stable-min}, HR_{stable-max}]$ we expect the user is currently in the stable stage. The value of $HR_{stable-min}$ and $HR_{stable-max}$ could be observed or calculated through experiments and data collection.
2. Secondly, we focus on the rising stage. When the heartbeat rate is smaller than $HR_{stable-min}$, the user is in rising stage. We use simple linear relationship to analyze the rising stage. The gradient of heartbeat rate with respect to time in the rising stage is represented as r. We can observe that the r value is not a constant. Therefore, we define the minimum value and maximum value of r for certain acceleration index to be r_{min} and r_{max} respectively. Similarly, we can observe and calculate the value of r_{min} and r_{max} from the data collection and observation.

3. The dropping stage is defined when the heartbeat rate is larger than the $HR_{stable-max}$. In this case the heartbeat rate decreases gradually until reaching to the stable stage. For simplicity, we also assume this relationship to be a linear. The gradient for the heartbeat rate with respect to time in the dropping stage is represented as d. Similarly, we define the minimum value and maximum value of d for certain acceleration index to be d_{min} and d_{max}.
4. All the above constants, i.e. $HR_{stable-min}$, $HR_{stable-max}$, r_{min}, r_{max} d_{min} and d_{max} are constants for a predefined acceleration index for a specific person. We are able to set their values for different people based on observations and measurements. All these values are similar to threshold. If we need the system to be more sensitive, we can set the interval smaller. If we want the system to be less sensitive, we can set the interval to be slightly bigger.
5. For example, from the Figure 6-14 above, we can observe point A to point B is the rising stage for acceleration index roughly equal to 130. According to the chart between A and B, we may decide that the r_{min}, r_{max} are equal to 0.2 and 0.5 respectively.
6. After we get the value for $HR_{stable-min}$, $HR_{stable-max}$, r_{min}, r_{max} d_{min} and d_{max}, we can make use of the values to implement the alert algorithm.

6.5.2 Terminology

1. We define the *deviation* between a real number x and a real interval [a, b] as following:

If x < a, the deviation is calculated as a-x.

If x > b, the deviation is calculated as x-b.

If a <=x <=b, the deviation is defined as 0

2. Notation min {a, b} is defined to be the minimum value between a and b.
3. Notation max {a, b} is defined to be the maximum value between a and b.

6.5.3 Algorithm description

1. We need measure the sensor data periodically. We use AI to denote the acceleration index, and HR to indicate the heartbeat rate. We use k = 1, 2, 3… to indicate each measurement. Hence, AI(k) and HR(k) indicate the acceleration index and heartbeat rate in kth measurement respectively. The time interval between kth measurement and (k-1)th measurement is defined as $\Delta t(k)$. Usually $\Delta t(k)$ must be very small.
2. We calculate the error detected for kth measurement based on the (k-1)th measurement. The acceleration index and heartbeat rate for (k-1)th measurement is represented as AI(k-1) and HR(k-1) respectively.
3. For (k-1)th measurement, based on the AI(k-1) we get the minimum value and maximum value of heartbeat in the stable stage, denoted by $\mathrm{HR}_{\mathrm{stable-min}}(k-1)$ and $\mathrm{HR}_{\mathrm{stable-max}}(k-1)$ respectively. The minimum value and maximum value of gradient in rising phase based on AI(k-1) may also be denoted as $\mathrm{r}_{min}(\mathrm{k}-1)$ and $\mathrm{r}_{max}(\mathrm{k}-1)$ The

minimum value and maximum value of gradient in the dropping stage may be denoted as $d_{min}(\mathrm{k}-1)$ and $d_{max}(\mathrm{k}-1)$.

4. Based on the known values of HR(k-1), $\mathrm{HR}_{\mathrm{stable-min}}(k-1)$, $\mathrm{HR}_{\mathrm{stable-max}}(k-1)$, $\mathrm{r}_{min}(\mathrm{k}-1)$, $\mathrm{r}_{max}(\mathrm{k}-1)$, $d_{min}(\mathrm{k}-1)$ and $d_{max}(\mathrm{k}-1)$, we can calculate a valid interval for expected value of heartbeat rate of the next measurement HR(k). We compare the expected value of HR(k) and real measured value of HR(k). There are three different scenarios.
5. If HR(k-1) < $\mathrm{HR}_{\mathrm{stable-min}}(k-1)$, it means at the (k-1)th measurement, the user is still in the rising stage. Since we expect a linear relationship between the heartbeat rate with respect to time in rising stage, we calculate the expected value of HR(k) based on HR(k-1) as follows:

$$\mathrm{Expect}\big(\mathrm{HR(k)}\big) = HR(k-1) + r * \Delta t(k)$$

Therefore, the expected minimum and maximum value of HR(k) can be defined as:

$$\mathrm{Expect}\big(\mathrm{HR(k)}\big)_{min} = HR(k-1) + r_{min} * \Delta t(k)$$

$$\mathrm{Expect}\big(\mathrm{HR(k)}\big)_{max} = HR(k-1) + r_{max} * \Delta t(k)$$

Moreover, expected value of HR(k) should not exceed the maximum stable value, $\mathrm{HR}_{\mathrm{stable-max}}(k-1)$. Therefore, we expect HR(k) is inside the interval

$$[\text{Expect}(\text{HR(k)})_{min}, \min\{\text{Expect}(\text{HR(k)})_{max}, \text{HR}_{\text{stable}-\max}(k-1)\}]$$

Equation 1

Therefore, we can calculate the error for this kth measurement E(k) to be the deviation between real measured value of HR(k) and the interval described in Equation 1.

6. If HR(k-1) is inside the interval

$$[\text{HR}_{\text{stable}-\min}(k-1), \text{HR}_{\text{stable}-\max}(k-1)]$$

Equation 2

It means at the (k-1)th measurement, the user has already in the steady stage. Therefore, we expect the value of HR(k) is still in the interval of Equation 2. In this case, the error for this kth measurement E(k) can be calculated as the deviation between the real measured value of HR(k) and the interval described in Equation 2.

7. If $\text{HR(k}-1) > \text{HR}_{\text{stable}-\max}(k-1)$, it means the user is in dropping stage. We can calculate the expected interval by the gradient in the dropping phase d. Therefore, the expected minimum and maximum value of HR(k) can be defined as:

$$\text{Expect}(\text{HR(k)})_{min} = HR(k-1) + d_{min} * \Delta t(k)$$

$$\text{Expect}(\text{HR(k)})_{max} = HR(k-1) + d_{max} * \Delta t(k)$$

Moreover, the expected value of HR(k) should not smaller than the minimum stable value $\mathrm{HR}_{\mathrm{stable-min}}(k-1)$. Hence the error for the kth measurement can be defined as the deviation between real measured value of HR(k) and the interval

$$[\max\{\mathrm{Expect}\big(\mathrm{HR(k)}\big)_{min}, \mathrm{HR}_{\mathrm{stable-min}}(k-1)\},$$
$$\mathrm{Expect}\big(\mathrm{HR(k)}\big)_{max}, \}]$$

8. All the previous procedures are able to calculate the error for the kth measurement E(k). We need make use of the error in each measurement to get a cumulative error. And we define the cumulative error at the kth measurement as followed:

$$\mathrm{E}_{\mathrm{cumulative}}(k) = \alpha * E_{cumulative}(k-1) + (1-\alpha) * E(k).$$

In the formula, α is the likelihood of the last step cumulative error. And $(1-\alpha)$ is the likelihood of E(k). α is a predefined constant. We define initial value of cumulative error $\mathrm{E}_{cumulative}(0) = 0$.

9. Last but not least, a threshold is set for cumulative error, which is denoted by $\mathrm{E}_{\mathrm{threshold}}$.Whenever $\mathrm{E}_{\mathrm{cumultive}}(k) > E_{threshold}$, the cumulative error is too big and an alert need to be shown to remind the user.

6.5.4 Algorithm test

We have done the algorithm test on the real measurement. Firstly, we observed and decided the values for r_{min}, r_{max}, $\mathrm{HR}_{\mathrm{stable-min}}$, $\mathrm{HR}_{\mathrm{stable-max}}$, d_{min} , and

d_{max}. We measured these values for some key values of acceleration index. Based on the data for the key values of the acceleration index, we calculate their value under other values of the acceleration index.

Acceleration index	r_{min}	r_{max},	$HR_{stable-min}$,	$HR_{stable-max}$,	d_{min}	d_{max}
1.422	0	0.21	80	120	-0.4	-0.2
5.28	0	0.21	80	120	-0.4	-0.2
7.61	0.13	0.21	100	120	-0.35	-0.2
8.88	0.23	0.34	110	130	-0.3	-0.1
14.5	0.6	0.9	170	190	-0.2	-0
22.23	0.6	0.9	170	190	-0.2	-0

Table 6 Values of the constants under certain key values of acceleration index

We collected value of the constants for six key values of acceleration index, shown in Table 6. If we need to find the value of the constants for another value of acceleration index, we can calculate the data proportionally.

For example, assume we want to calculate the r_{max} value when acceleration index is equal to 8. Since 8 is between the interval of [7.61, 8.88], we need to focus row 3 and row 4 in the table. We calculate the r_{max} proportionally as following:

$$r_{max}(acceleration\ index = 8) = \frac{(0.34 - 0.21)}{(8.88 - 7.61)} * (8 - 7.61) + 0.21 = 0.25$$

Though the value calculated by the above algorithm is an estimated value, it is accurate enough in the algorithm. Moreover, we choose the likelihood value and error threshold as follows:

$$\alpha = 0.9$$

$$\mathrm{E}(\text{threshold}) = 2$$

In order to test the accuracy of the algorithm, we need to run this algorithm in different scenarios for comparison. We did not have the real data from people with abnormal heartbeat rate, therefore, we decided to add a Gaussian noise to the normal heartbeat rate data to make the heartbeat rate to be abnormal. Therefore, I have run this algorithm on 2 different scenarios.

1. In case one, we process the real data mentioned in the Figure 6-14.
2. In case two, we add a Gaussian noise to the heartbeat rate in case one, and process the data with the Gaussian noise. We choose the Gaussian noise with mean equal to 0, and deviation equal to 8.

We compare the output for the two cases. Since case one is the real data collected from healthy people, while case two contains a Gaussian noise to its heartbeat rate data, we expect a higher cumulative error output from the case two compared to case one. The output figures for these two cases are shown in the Figure 6-15 and Figure 6-16 respectively.

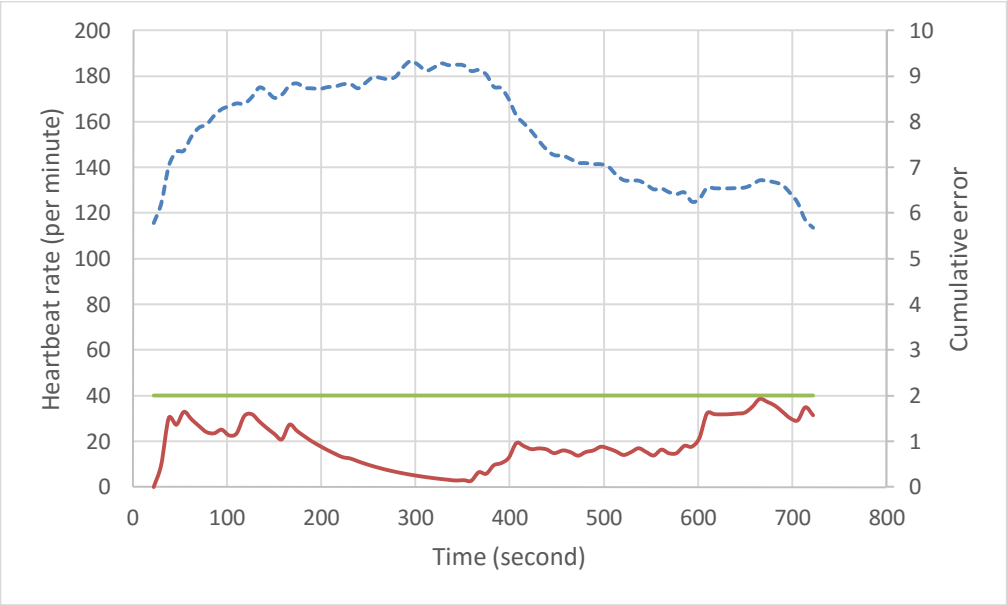

Figure 6-15 Heartbeat and cumulative error for real measurement. The dotted line is the heartbeat rate, the solid line is the cumulative error generated using our algorithm. The horizontal line is the threshold.

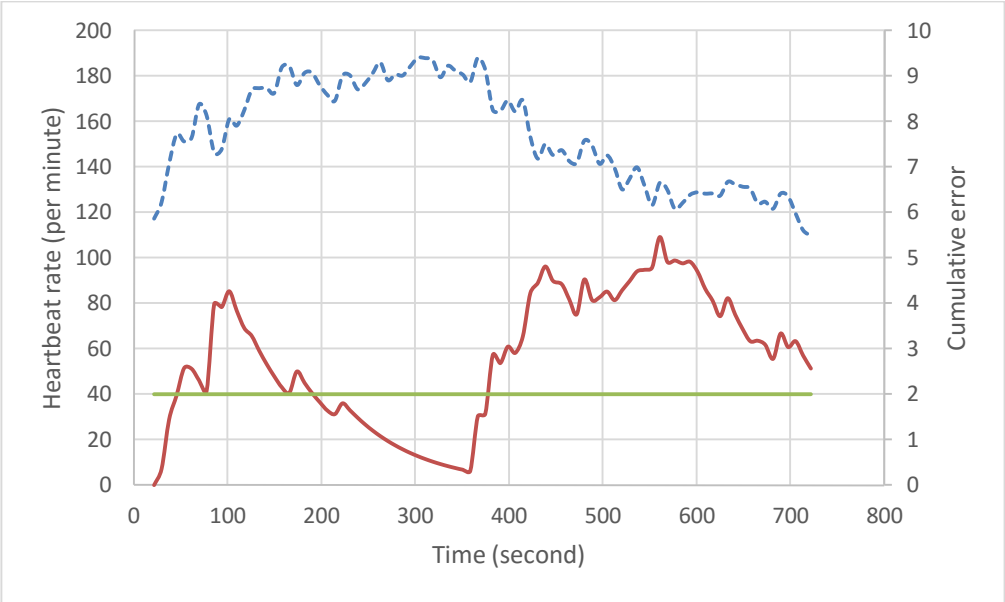

Figure 6-16 Heartbeat and cumulative error when adding a Gaussian noise to the heartbeat. The dotted line is the heartbeat rate added with Gaussian noise. The solid line is the cumulative noise we get using our algorithm. The horizontal line is threshold.

The Figure 6-15 and Figure 6-16 show the results after we have executed the algorithm. In both of the charts, the dotted line indicates heartbeat rate, which based on the scale on the left. The solid line indicates the cumulative error output from our algorithm, which based on the scale on the right. The horizontal line is the threshold. Both of these two charts use the same vertical axis scale. We have several observations regarding to the result.

1. Firstly, in case one, the cumulative error we calculated is mostly below the threshold, which means we do not receive too many alerts.
2. In case two, the heartbeat line fluctuates much deeper. Since we add a Gaussian noise, the cumulative error increases, compared to when there is no Gaussian noise in case one.

3. In case two, we get a much higher cumulative error at rising stage and dropping stage, while in the stable stage, we get a much lower cumulative error, because in our model and observation, we do expect a fluctuating in the stable stage.

In conclusion, we introduce and develop a simple while useful abnormal detection algorithm. We have tested this algorithm based on two cases, one is the real measured data, and the other one is the real measured data adding a Gaussian noise. After added a Gaussian noise, the data generates a much higher cumulative error. The alert algorithm can help us in the abnormal detection. Based on this algorithm, we can help the users to understand abnormal conditions. In the future, we can easily implement this algorithm on the mobile device, therefore, the system is able to generate alert to the users in real time, when large cumulative error is detected by the system.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Thesis Contribution

In this project we have built an ECG monitoring system based on mobile platforms. We have used this system to collect sensor data and have done some analysis to the data.

1. We have developed an ECG monitoring system based on mobile platform. The system is able to operate standalone without Internet connection. It contains two modes: gateway mode and server mode for different usage scenarios.
2. We have implemented some algorithms to process the ECG sensor data, therefore the system can remove some of the noise and detect the R peak in real time in the mobile device.
3. We have also built the system to record down accelerometer sensor data at the same time. We have collected ECG sensor data and accelerometer sensor data using the system. Based on the sensor data we collected, we have analyzed the relationship between the ECG data and accelerometer data.
4. Based on the data we have collected, we have also developed an algorithm to generate alerts based on the ECG sensor data and accelerometer sensor data.

7.2 Future work

In my opinion, we still have some future studies for this project.

1. Firstly, we need build more sensors inside the system, such as temperature sensors. More sensors mean we could collect more data and get more information from the users.
2. Regarding the heartbeat rate and acceleration index, we need to collect more data from various people. We need more data to analyze a more accurate and complex relationship between sensor data.
3. We need to investigate relationship among various sensors, therefore, we can develop a better algorithm based on various sensors data together to help the user know his health condition.

BIBLIOGRAPHY

[1] Z. Zhaoyang, "ECG-Cryptography and Authentication in Body Area Networks," in *Information Technology in Biomedicine*, 2012.

[2] United States Centers for Disease Control and Prevention, "Heart Disease Facts," 19 February 2015. [Online]. Available: http://www.cdc.gov/heartdisease/facts.htm. [Accessed 1 May 2015].

[3] M. Kenney and B. Pon, "Structuring the Smarthone Industry: Is the Mobile Internet OS Platform the Key?," *Journal of Industry, Competition and Trade,* vol. 11, no. 3, pp. 239-261, 2011.

[4] FONPIT, "Smartphone Evolution: this is the Google Nexus phone series," 2015. [Online]. Available: http://www.androidpit.com/google-nexus-evolution. [Accessed 2015].

[5] E. Stankevich and I. Paramonov, "Using Bluetooth on Android Platform for mHealth Development," in *Proceedings of the 10th Conference of FRUCT Association*, Yaroslavl, Russia, 2012.

[6] J. M. Cano-Garcia, E. Gonzalez-Parada, V. Alaron-Collantes and E. Casilari-Perez, "A PDA-based portable wireless ECG monitor for medical personal area networks," in *IEEE Melecon*, Benalmandena(Malaga), Spain, 2006.

[7] A. Al-Omary, W. EI-Mdany and R. Al-Hakim, "Heart Disease Monitoring System Using Web and Smartphone," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 3, no. 4, p. 8265, 2014.

[8] Z. Fang and D. Lai, "Uninterrupted ECG Mobile Monitoring," *International Journal of Bioelectromagnetism*, vol. 9, no. 1, p. 33, 2007.

[9] S. Kang, S. Kwon, C. Yoo, S. Seo, K. Park, J. Song and Y. Lee, "Sinabro: Opportunistic and Unobtrusive Mobile Electrocardiogram Monitoring System," in *15th International Workshop on Mobile Computing Systems and Applications (HotMobile)*, Singapore, 2014.

[10] R. Gupta, H. Chatterjee and M. Mitra, "An online ECG QRS Detection Technique," in *ACEEE*, 2012.

[11] Medical Training and Simulation LLC, "ECG Interpretation," 2015. [Online]. Available: http://www.practicalclinicalskills.com/ecg-interpretation.aspx. [Accessed May 2015].

[12] "How to read an EKG," 2 October 2011. [Online]. Available: http://www.todayifoundout.com/index.php/2011/10/how-to-read-an-ekg-electrocardiograph/. [Accessed April 2014].

[13] Healthwise, "Electrocardiogram (EKG) Components and Intervals," 12 March 2014. [Online]. Available: http://www.webmd.com/heart/ekg-components-and-intervals. [Accessed May 2015].

[14] National Heart, Lung, and Blood Institute, "What Does an Electrocardiogram Show?," 1 October 2010. [Online]. Available: http://www.nhlbi.nih.gov/health/health-topics/topics/ekg/show. [Accessed May 2015].

[15] Apple Inc., "MFi Program," 2015. [Online]. Available: https://developer.apple.com/programs/mfi/. [Accessed 2015].

[16] IDC, "Android and iOS Continue to Dominate the Worldwide Smartphone Market with Android Shipments Just Shy of 800 Million in 2013, According to IDC," 12 Feb 2014. [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS24676414. [Accessed May 2015].

[17] "Android Developers," Google, 2015. [Online]. Available: https://source.android.com/. [Accessed 14 April 2015].

[18] "IDC: Smartphone OS Market share," IDC, 2015. [Online]. Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp. [Accessed 14 April 2015].

[19] M. Rosoff, "9 Charts That Show The Amazing Growth Of Android," 26 March 2012. [Online]. Available: http://www.businessinsider.com/android-9-charts-that-show-its-amazing-growth-2012-3. [Accessed May 10 2015].

[20] F. William and T. Carol, "Software Reuse: Metrics and Models," *ACM Computing Surveys,* vol. 28, pp. 416-435, 1996.

[21] Clearbridge Vitalsigns Pte Ltd., 2014. [Online]. Available: http://www.clearbridgevitalsigns.com/fit.html. [Accessed 12 April 2015].

[22] Raspberry Pi Foundation, "What is a Raspberry Pi," Raspberry Pi Foundation, [Online]. Available: http://www.raspberrypi.org/help/what-is-a-raspberry-pi/. [Accessed 6 April 2015].

[23] Python Software Foundation, "Compare Python to other languages," 1997. [Online]. Available: https://www.python.org/doc/essays/comparisons/. [Accessed 2015].

[24] Kaazing, "What is WebSocket?," 2013. [Online]. Available: https://www.websocket.org/. [Accessed May 2015].

[25] A. Q. Bhat, "Baseline wander Removal in ECG using an efficient method of EMD in combination with wavelet," *IOSR Journal of VLSI and Signal Processing,* vol. 4, no. 2, p. 76, April 2014.

[26] A. Fasano, V. Villani and L. Vollero, "Baseline Wander Estimation and Removal," in *33rd Annual International Conference of the IEEE EMBS*, Boston, 2011.

[27] H. Chen and S. Chen, "A Moving Average based Filtering System with its Application to Real-time QRS Detection," in *Computers in Cardiology*, 2003.

[28] E. Burns, "QRS Complex Morphology," 2015. [Online]. Available: http://lifeinthefastlane.com/ecg-library/basics/qrs-complex-morphology/. [Accessed November 2015].

[29] Google Inc, "SensorEvent," 2015. [Online]. Available: http://developer.android.com/reference/android/hardware/SensorEvent.html. [Accessed 10 June 2016].

[30] "SensorEvent," Google, 2015. [Online]. Available: http://developer.android.com/reference/android/hardware/SensorEvent.html. [Accessed 2015].

[31] Y. Yang, L. Ji, H. Wu and J. Wu, "An Exercise-Driven Heart Rate Statistical Process Model," in *Information Fusion (FUSION), 2012 15th International Conference on*, Singapore, 2012.

[32] Danbury Hospital, "Danbury Hospital Patient Education Sheet - Exercise for the Heart," 2015. [Online]. Available: http://www.danburyhospital.org/en/Patient-and-Visitor-Information/Information-Guides/~/media/Files/Patient%20Education/patiented-english/pdf_Cardiac/ExerciseforHeart.ashx. [Accessed November 2015].

[33] Google, "Saving Data in SQL Database," 2014. [Online]. Available: http://developer.android.com/training/basics/data-storage/databases.html. [Accessed 2015].

[34] AndroidPlot, 2015. [Online]. Available: http://androidplot.com/. [Accessed 2015].

[35] D. Bosomworth, "Mobile Marketing Statistic 2015," Smart Insights, 15 January 2015. [Online]. Available: http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/. [Accessed May 2015 2015].

[36] Google Inc., "SharedPrefernces | Android Developer," May 2015. [Online]. Available: http://developer.android.com/reference/android/content/SharedPreferences.html. [Accessed May 2015].

[37] Google Inc., "Saving Data in SQL Databases," 2015. [Online]. Available: http://developer.android.com/training/basics/data-storage/databases.html. [Accessed 2015].

[38] A. Boughton, "Object Relational Database Mapping," 2011. [Online]. Available: http://www.cs.colorado.edu/~kena/classes/5448/s11/presentations/ordmpresentation.pdf. [Accessed 2015].

[39] "OrmLite - Lightweight Object Relational Mapping (ORM) Java Package," 2014. [Online]. Available: http://ormlite.com/. [Accessed 2015].

[40] Play Framework Community, "Play Framework makes it easy to build web application with Java & Scala," 2015. [Online]. Available: https://www.playframework.com/. [Accessed May 2015].

[41] EPFL, "The Scala programming language," 2015. [Online]. Available: http://www.scala-lang.org/. [Accessed May 2015].

[42] S. Brent and G. Leon, "Scala and Go: A comparision of concurrency features," 2012. [Online]. Available: http://www.cs.colorado.edu/~kena/classes/5828/s12/presentation-materials/smithbrentgibsonleon.pdf. [Accessed 2015].

[43] MongoDB Inc., "NoSQL Database explained," 2015. [Online]. Available: https://www.mongodb.com/nosql-explained. [Accessed May 2015].

[44] Liris, "WebSocket client for Python," 2014. [Online]. Available: https://pypi.python.org/pypi/websocket-client/. [Accessed 2015].

[45] Google, "Activity," May 2015. [Online]. Available: http://developer.android.com/reference/android/app/Activity.html. [Accessed May 2015].

[46] Bluetooth Special Interest Group (SIG), "Introduction to Bluetooth Technology," Washington.

[47] Google Inc, "Services | Android Developers," 2015. [Online]. Available: http://developer.android.com/guide/components/services.html. [Accessed June 2015].

[48] Vitatech Electromagnetics, "EKG," 2015. [Online]. Available: http://www.vitatech.net/glossary/ekg/. [Accessed 2015 August].

[49] D. Rohmer, "A barebones WebSocket client and Server implemented written in 100% Java," August 2014. [Online]. Available: http://java-websocket.org/. [Accessed May 2015].